Learning Machine Monitoring Models from Sparse and Noisy Sensor Data Annotations

DISSERTATION

zur Erlangung des Grades eines Doktors

der Ingenieurwissenschaften

vorgelegt von

M. Sc. Christian Reich

geb. am 28.01.1984 in Karlsruhe

eingereicht bei der Naturwissenschaftlich-Technischen Fakultät

der Universität Siegen

Siegen 2019

Betreuer und erster Gutachter

Prof. Dr. Kristof Van Laerhoven

Universität Siegen

Zweiter Gutachter

Prof. Dr.-Ing. habil. Roman Obermaisser

Universität Siegen

Tag der mündlichen Prüfung

29. Juni 2020

Gedruckt auf alterungsbeständigem holz- und säurefreiem Papier

To my wonderful family
and beloved darling Silke

Abstract

Present-day requirements on efficiency and quality of manufacturing processes necessitate constant monitoring of machine tools and machining processes. Although automated, sensor-based machine monitoring techniques are described in literature, real-world production shops still exhibit a high degree of human intervention, which tends to be both expensive and error-prone. This is due to three challenges that such machine monitoring systems are confronted with which this thesis will address:

First of all, long-term deployable systems require robust predictive models. The models need to generalize across user-initiated adjustments of process parameters and changes of workpiece types, such that trained models still match the distribution of newly incoming test data by independence of *covariate shift* among training and testing data distributions. The variance in sensor data is often more influenced by such parameter adjustments and workpiece changes than by actual anomalies. This dominance of covariate shift over class-discriminative information in sensor data is challenging. Secondly, most performant predictive models are (semi-)supervised, requiring large sets of labeled sensor data. Annotation of anomalous data is expensive and comes with a severe risk of machine damages when deliberately provoking anomalies. Finally, high-performant models rely on high memory resources, long training and model execution times or specific hardware for training (e.g., GPUs). These requirements conflict with the desire of companies for industrially robust and compact embedded sensor systems and short model execution times (allowing timely alerts of and quick responses to potentially critical anomalies). Evaluation directly on embedded sensor systems allows for an increased data security, compactness allows to retrospectively equip machines with these sensor systems.

The first part of the thesis is concerned with defining features tailor-made for specific machine monitoring tasks that generalize across covariate shift. To that end, domain expertise about machine and process characteristics is included in custom-built preprocessing models for segmentation of sensor data and tracking of discrete frequency components. The second part of the thesis focuses on low-cost annotation and detection of "in-the-wild" recorded anomalies. A prototypical evaluation system was developed specifically for harsh industrial environments and deployed there. The system enables data recording, on-system evaluation and reporting of potential anomalies, both supporting operators in decision processes and allowing for annotation of sensor data by operators' feedback to anomaly propositions. Evaluations of this prototypical system and the resulting data suggested that involved anomaly detection models were overly simple, so more sophisticated unsupervised neural anomaly detection models were tested. In addition, two semi-supervised extensions trained with expert labels and automatically generated, therefore weak labels were compared. Both unsupervised and semi-supervised neural anomaly detectors prove to be well-suited, generalizing across several weeks of data exhibiting covariate shift. All presented methods respect constraints imposed by embedded systems used for machine monitoring and the need of timely responses to anomalies.

Zusammenfassung

Heutige Anforderungen an Effizienz und Qualität von Fertigungsprozessen bedingen eine ständige Überwachung von Werkzeugmaschinen und Bearbeitungsprozessen. Obwohl automatisierte, sensorbasierte Maschinenüberwachungssysteme in der Literatur bestehen, sind reale Produktionsstätten weiterhin durch einen hohen Anteil menschlicher Beteiligung gekennzeichnet, was teuer und fehleranfällig zu sein tendiert. Dies ist durch drei Herausforderungen bedingt, denen sich solche Überwachungssysteme ausgesetzt sehen und welche in dieser Arbeit adressiert werden:

Zunächst einmal erfordern Überwachungssysteme im Langzeitbetrieb robuste, prädiktive Modelle. Die Modelle müssen über nutzerbedingte Anpassungen von Prozessparametern und Wechsel von Werkstücktypen hinweg generalisieren, sodass trainierte Modelle weiterhin zur Verteilung neu aufgenommener Testdaten passen und damit unabhängig gegenüber des *Covariate Shifts* zwischen Trainings- und Testdatenverteilungen sind. Die Varianz in Sensordaten ist häufig stärker von derartigen Parameteranpassungen und Werkstücktypwechseln beeinflusst als von tatsächlichen Anomalien. Diese Dominanz des Covariate Shifts gegenüber signalklassenunterscheidender Information in Sensordaten ist herausfordernd. Zweitens sind die leistungsstärksten prädiktiven Modelle (teil)überwacht und erfordern große Mengen annotierter Sensordaten. Die Annotation von Anomalien in Sensordaten ist teuer und bedingt mögliche Folgeschäden an den Maschinen, wenn die Anomalien bewusst herbeigeführt werden. Zu guter Letzt benötigen die leistungsfähigsten Modelle viel Speicherplatz, lange Trainings- und Modellausführungszeiten oder spezielle Hardware wie GPUs im Trainingsprozess. Diese Anforderungen stehen im Widerspruch zu fertigungsbetrieblichen Wünschen nach industriell robusten, kompakten eingebetteten Sensorsystemen und kurzen Modellausführungszeiten, welche sowohl zeitnahe Warnungen von als auch schnelle Reaktion auf potentiell kritische Anomalien ermöglichen. Die Datenauswertung innerhalb des eingebetteten Sensorsystems ermöglicht eine höhere Datensicherheit, die Kompaktheit des Systems erlaubt nachträgliche Ausstattung von Maschinen mit diesen Sensorsystemen.

Der erste Teil der Arbeit umfasst die Definition maßgeschneiderter Merkmale, welche über den Covariate Shift der Daten hinweg generalisieren, für spezifische Maschinenüberwachungsaufgaben. Dafür wird Fachexpertenwissen über Maschinen- und Prozesscharakteristika in maßgeschneiderten Vorverarbeitungsmodellen zur Segmentierung von Sensordaten und Nachverfolgung diskreter Frequenzkomponenten abgebildet. Der zweite Teil konzentriert sich auf die kostengünstige Annotation und Erkennung von im realen Bearbeitungsprozess aufgenommenen Anomalien. Ein prototypisches Auswertungssystem, welches speziell auf raue industrielle Umgebungen abgestimmt ist, wurde entwickelt und in einer solchen Umgebung eingesetzt. Dieses System ermöglicht Datenaufnahme, Datenauswertung im System und Meldung potentieller Anomalien. Damit unterstützt es sowohl Maschinenbediener in Entscheidungsprozessen und ermöglicht eine Annotation der Sensordaten über die Rückmeldung der Bediener auf gemeldete Anomalien. Auswertungen dieses prototypischen Systems und der resultieren-

den Daten legten eine allzu simple Art der eingesetzten Anomalieerkennungsmodelle nahe, weswegen ausgefeiltere, unüberwachte neuronale Anomalieerkennungsmodelle getestet wurden. Zusätzlich wurden zwei teilüberwachte Modellerweiterungen mit Expertenlabels sowie automatisch generierten und damit schwächeren Labels trainiert. Sowohl die unüberwachten als auch die teilüberwachten neuronalen Anomalieerkennungsmodelle bewiesen sich als geeignet, sie generalisierten über einige Wochen von Daten und den auftretenden Covariate Shift hinweg. Alle vorgestellten Methoden beachten die Einschränkungen, welche durch die für die Maschinenüberwachung eingesetzten eingebetteten Systeme und den Bedarf an zeitnahen Reaktionen gegenüber Anomalien bedingt sind.

Acknowledgments

Although a PhD thesis aims at proofing scientific independence of the author, I am happy to be able to distinguish this from scientific loneliness. This is due to the support of a wide range of people.

First and foremost, I want to express sincere gratitude to my thesis supervisor, Prof. Dr. Kristof Van Laerhoven, for scientific guidance and the possibility to discuss my ideas at multiple occasions both with himself and other PhD students from his research group in regular seminar meetings. In addition, I want to thank Prof. Dr. Van Laerhoven for his open-minded and motivational kindness, that always gave confidence when feeling stuck with a scientific problem. I also want to thank Prof. Dr.-Ing. habil. Roman Obermaisser for willing to be the co-examiner of this thesis.

This thesis was financed by the Robert Bosch GmbH and mainly conducted at the facilities of Bosch's "Corporate Sector Research and Advance Engineering (CR)" in Renningen. I would first like to thank Andre Kretschmann, my department leader, for accepting me as PhD candidate in the "Microsystems and Nanotechnologies (CR/ARY)" department and financing this thesis. I would also like to thank my group leader, Stefan Leidich, for the frequent discussions of my PhD topic especially in the beginning of my time as doctoral student, and for connecting me with other Bosch employees for discussion. Sincere gratitude is expressed also to all members of the research project I worked in: Max Schellenberg for support with measurement setups, Philip Jung for software discussions, and most importantly Ahmad Mansour, my supervisor at Bosch, for frequent discussions in our regular meetings and scientific guidance. In addition, thanks to Ricardo Ehrenpfordt, my former project leader, for connecting me with people inside and outside Bosch working on related topics. And finally, thanks to my PhD colleagues both at the Ubiquitous Computing research group in Siegen and in Renningen, both for scientific discussion and for sometimes necessary distraction from scientific topics.

Furthermore, I would like to thank my parents Werner and Ursula and my sister Natalie for constant support, motivation and love, helping me to never feel alone on my PhD journey. Finally, I want to thank Silke, my faithful companion, not only for her unconditional love but also for the extraordinary strength and endurance she shows in her life and which has always been an inspiration to me regarding my own work.

Renningen, November 2019
Christian Jörn Reich

Contents

Notation

Mathematical symbols

Notation	Meaning
$\mathbb{N}$	Set of natural numbers
$\mathbb{R}$	Set of real numbers
$\mathbb{C}$	Set of complex numbers
$\mathbf{M}$	Matrices (bold font, capital)
$\mathbf{v}$	Vectors (bold font, lowercase)
s	Scalars (thin font, lowercase)
$\int$	Integral operator
$(\cdot)'$	Differentiation operator
$\frac{d(\cdot)}{dt}$	Differentiation with respect to time
$\frac{\partial(\cdot)}{\partial t}$	Partial differentiation with respect to time
$\langle \cdot, \cdot \rangle$	Inner product
$\overline{(\cdot)}$	Complex conjugation operator
$(\cdot)^H$	Hermitian operator
$(\cdot)^+$	Pseudoinversion operator
$\|\cdot\|_2$	Norm operator (here: ℓ_2 norm)
$*$	Convolution operator
$\odot$	Element-wise (Hadamard) product
$D(\cdot \| \cdot)$	Divergence operator
$\mathbb{E}[\cdot]$	Expectation operator
$\sim$	Sampling operator
$\propto$	Proportionality operator
$\triangleq$	Definition operator
$\subseteq$	Subset operator
$\setminus$	(Absolute) complement of a set

Abbreviations (1–F)

Abbreviation	Meaning
1NN	1-nearest neighbor
AE	Autoencoder
AIC	Akaike information criterion
ANN	Artificial neural network
AR	Autoregressive
ARMA	Autoregressive moving average
ARV	Average rectified values
AUC	Area under curve
AWGN	Additive white Gaussian noise
BIC	Bayesian information criterion
BOCPD	Bayesian online changepoint detection
CBM	Condition-based maintenance
CEC	Centerless external cylindrical
CM	Condition monitoring
CNN	Convolutional neural network
CPRD	Changepoint recurrence distribution
PCA	Compressive sensing
CSD	Cumulative squared distance
CUSUM	Cumulative sum control chart
CV	Cross-validation
DBA	DTW barycenter averaging
DDM	Distribution derivative method
DFC	Discrete frequency component
DFT	Discrete Fourier transform
DGM	Data-generating mechanism
DNN	Deep neural network
DOF	Degree of freedom
DTW	Dynamic time warping
DWT	Discrete wavelet transform
ECG	Electrocardiogram
ED	Euclidean distance
EM	Expectation maximization
ESPRIT	Estimation of Signal Parameters via Rotational Invariance Techniques
FCN	Fully convolutional network
FD	Frequency domain
FFT	Fast Fourier transform
FN	False negative
FNR	False negative rate
FP	False positive
FPR	False positive rate
FSM	Finite state machine

Abbreviations (G–P)

Abbreviation	Meaning
GAN	Generative adversial network
GBI	Generalized bayesian inference
GLR	Generalized likelihood ratio
GMM	Gaussian Mixture Model
GP	Gaussian Process
GPR	Gaussian process regression
GPU	Graphics processing unit
GRU	Gated recurrent unit
GUI	Graphical user interface
HHT	Hilbert-Huang transform
HMI	Human-machine interface
HMM	Hidden Markov model
HT	Hough transform
HW	Hardware
IID	Independent and identically distributed
KDE	Kernel density estimator
KL	Kullback-Leibler
KLIEP	Kullback–Leibler Importance Estimation Procedure
KNN	k-nearest neighbors
L2R	Left-to-right
LF	Labeling function
LSE	Line spectral estimation
LSTM	Long short-term memory
MA	Moving average
MAP	Maximum a posteriori
MEMS	Microelectromechanical systems
MFCC	Mel-frequency cepstral coefficient
HMM	Machine health monitoring
MLP	Multilayer perceptron
MRP	Markov renewal process
MSE	Mean squared error
MUSIC	Multiple signal classification
NC	Nearest centroid
NMF	Nonnegative matrix factorization
PCA	Principal component analysis
PCCF	Predictive change confidence function
PE	Pearson
PGM	Probabilistic graphical model
PLA	Piecewise linear approximation

Abbreviations (R–W)

Abbreviation	Meaning
RELU	Rectified linear unit
RMS	Root mean square
RMSE	Root mean squared error
RNN	Recurrent neural network
ROC	Receiver operating characteristic
ROI	Region of interest
RUL	Remaining useful lifetime
RULSIF	Relative uLSIF
SAD	Semi-supervised Anomaly Detection
SDTW	Soft-DTW
SGD	Stochastic gradient descent
SNR	Signal-to-noise ratio
SOM	Self-organizing map
SSA	Singular spectrum analysis
STACS	Simultaneous Temporal and Contextual Splitting
STFT	Short-time Fourier transform
SVDD	Support Vector Data Description
SVM	Support vector machine
TCM	Tool condition monitoring
TCN	Temporal convolutional network
TD	Time domain
TF	Time-frequency
TFD	Time-frequency distribution
TSC	Time series classification
ULSIF	Unconstrained Least-Squares Importance Fitting
VAE	Variational autoencoder
VALSE	Variational line spectral estimation
WVT	Wigner-Ville transform

1

Introduction

Today's efficiency demands in modern factory workshops require permanent monitoring of both the production efficiency and health state of the production machines. This monitoring involves evaluation of data gathered via sensors. Apart from image- and video-based modalities, sensors attached to machine key points of interest have become state of the art in machine monitoring. Among most popular sensor types are acoustic emission, force and vibration sensors [252].

This thesis is situated in monitoring of machine tools by evaluation of data recorded with vibration and acceleration sensors attached to such machines. Monitoring of machine health and production efficiency based on these sensor measurements involves various steps [47]: Raw sensor data is typically preprocessed in order to get rid of signal artifacts or emphasize the relevant information in the time series. This is typically approached by filter methods, which detect and filter out discrete spurious frequency components or shape the relevant signal information by (multiple) broad-band filters. Afterwards, time series are segmented into subregions. Depending on the application, segmentation can be motivated by cutting out subregions of interest (e.g., segments of data containing the phenomena one wants to detect) or finding comparable regions in a recurrent stream of data (e.g., in machine tool applications, as outlined in the following section). After detecting the most relevant regions in frequency domain and time domain by applying preprocessing and segmentation techniques, characteristics of the data that help to detect the phenomena of interest have to be defined. In the pattern recognition community, these data characteristics are often referred to as *features*. In addition to emphasizing relevant and discriminative signal information, feature extraction often comes with a drastic concentration of information content, thus allowing downsampling of the data. Such features can be either handcrafted by human experts or learned directly from the data. Finally, the features can be used as information fed to a classifier model which decides whether the phenomena

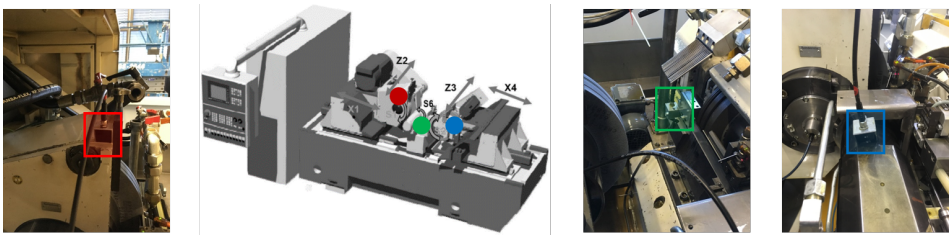


Figure 1.1: *Typical measurement setup considered for the data evaluations in this thesis. MEMS sensors are attached to key points of interest at the machine tool (colored points in second figure), most commonly the grinding wheel housing (red point in middle figure, first figure), workpiece support (green point in second figure, third figure) and dressing tool (blue point in second figure, fourth figure). These MEMS sensors are connected by a gateway system (i.e., an embedded computer) which is capable of performing data processing and connectivity tasks. Second subfigure adapted from [97].*

of interest are apparent in the current segment of the data. Such classifiers can either yield a binary decision (phenomena present? → yes/no) or multi-class decision (which of the different types of phenomena of interest is present?). When the phenomena of interest are assumed to be highly under-represented in the data compared to other signal classes, one often refers to an *anomaly detection* (binary problem) or *anomaly classification* (multi-class problem).

Training of anomaly detection or classification models were historically subdivided into two categories. When the data used for training the model are accompanied by annotations (*labels*) of the signal class the model can be learned from the labeled examples. This setting is categorized as *supervised training/learning*. When no annotations are given, one has to rely on other properties of the training data. Common strategies include identifying clusters or high density regions of values in the training feature data. Such a scenario of inaccessibility of labels for the training data is referred to as *unsupervised training/learning*.

For detection of anomalies in the health state or production process of machines, finding sensible positions to which sensors are attached to is a key task in order to obtain an effective machine monitoring system. In addition, a mechanically solid attachment is of vital importance in order to avoid damping effects that would limit the effective bandwidth of the attached sensors. Thus, the sensor systems have to be screwed tightly to the machine parts of interest.

For so-called centerless external grinding machines which are in the focus of this thesis, typical positions are illustrated in Fig. 1.1. The positions cover the most important grinding machine parts involved in the process of machining workpieces. The process of machining workpieces is visually summarized in Fig. 1.2. The workpiece is situated between grinding wheel and control wheel on the workpiece support. The grinding wheel approaches the workpiece and starts machining of the workpiece. Workpiece support and control wheel decelerate the workpiece. This difference in velocity of grinding wheel and control wheel

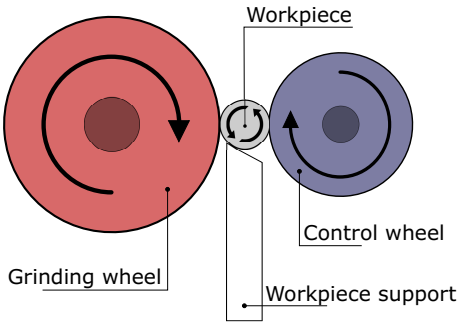


Figure 1.2: *Process of external cylindrical grinding. Figure inspired from [241].*

applies a force to the workpiece which induces the material removal. After a certain number of machined workpieces, the grinding wheel becomes dull and has to be sharpened again. This sharpening process is referred to as *dressing* and performed by a dressing tool/wheel.

1.1 Goal and Focus of Thesis

The overall goal of this thesis is developing algorithmic approaches for monitoring the health state and production efficiency of machine tools via embedded sensor systems. For this, data from sensors attached to key points of the machine tools are processed. Throughout the course of this thesis, multiple data sets were recorded. Each data set consists of multiple data records. Data records consist of sensor signals recorded for the duration of machining a single workpiece. In this thesis, trigger signals from the machine control program were used for subdividing sensor streams into successive data records. Each of the sensor signals in a data record illustrates multiple segments. The segments are due to the sequence of processing steps applied during the machining of successive workpieces with a profiled grinding wheel. Thus, segment borders depict the different stages of machining a single workpiece. Finally, each segment consists of multiple blocks with fixed length (e.g., 1024 raw data samples). The terminology is visually illustrated in Fig. 1.3 for a clear and concise understanding of the terms data set, data record, segment and block which are frequently used in later chapters.

The overall goal of this thesis can be located more clearly regarding different aspects:

- Regarding machine tool types, an emphasis is put on grinding machines. Grinding machines cover a large fraction of machine tools found in modern factory workshops. The reason for this is the variety of different workpiece geometries that can be processed by grinding machines and their high productivity. The latter is especially true for centerless grinding machines.
- Algorithmic approaches are researched for different predictive tasks. In

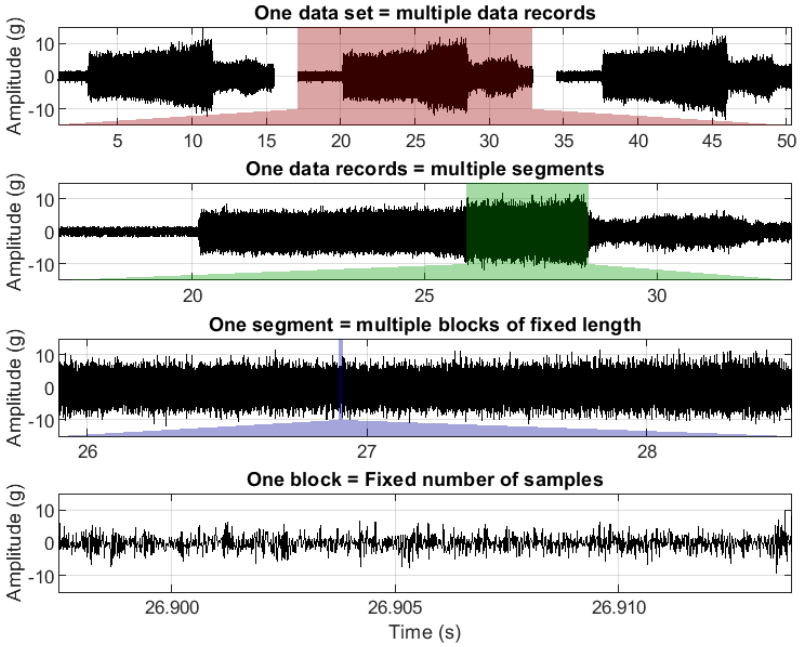


Figure 1.3: *Hierarchical structure of data considered in this thesis. Each recorded data set consists of multiple data records. These records in turn contain sensor signals with multiple successive segments. Each segment in turn consists of multiple block of fixed length (e.g., 1024 raw data samples).*

part I of this thesis, several specific monitoring tasks are considered. The most important ones are tool condition monitoring and detection of imbalances in rotating machinery. In part II of this thesis, generic (binary) anomaly detection approaches focusing on detecting deviations from the machine's normal (health or production) state are discussed rather than specific tasks as in part I.

- Finally, the focus of sensor types is set on microelectromechanical systems (MEMS) vibration and acceleration sensors. Compared to industrially established acoustic emission sensors, MEMS sensors have a smaller sampling rate and are thus expected to have lower energy demands but have a limited bandwidth in comparison. In addition, MEMS sensors are typically cheaper and smaller than acoustic emission sensors. Each of the sensor nodes illustrated in Fig. 1.1 contains both a MEMS single-axis vibration sensor and a MEMS tri-axial acceleration sensor. All attached sensor nodes are connected by a gateway system (i.e., an embedded computer). This gateway system has multiple purposes. Firstly, its major task is to connect the sensors to a data recording device. For measurements conducted for this

Table 1.1: *Technical data of the two sensor types used throughout this thesis*

Sensor type	Sampling rate	Sensitivity	Axes
Acceleration	2 kHz	±2 g at 12 bit	3
Vibration	62.5 kHz	±96 g at 10 bit	1

thesis, this was either a portable measurement case (used for recording of data evaluated in part I of this thesis) or a self-developed signal visualization and labeling prototype attached to the outside of machine tools (used for recording of data evaluated in part II of this thesis). Secondly, the gateway system exhibits more computational resources than the sensor nodes, allowing for simple, edge-based data processing tasks.

Technical characteristics of the MEMS vibration and acceleration sensors used in this thesis are listed in Table 1.1. The vibration sensor has a higher bandwidth than the acceleration sensor and is used throughout most chapters in this thesis, as many anomaly types manifest in high-frequency bands. Only in Chapter 4, measurements from acceleration sensors are evaluated, as the discrete frequency components that shall be detected are located in the bandwidth of 1 kHz provided by the acceleration sensors. Furthermore, the sensitivity of these acceleration sensors is higher, which is important for the methods discussed in Chapter 4.

1.2 Challenges of Thesis

Although automated, sensor-based monitoring of the machine's health state and production efficiency is highly relevant for production companies and such approaches exist in literature, real-world production shops still exhibit a high degree of human intervention and lack of automated machine monitoring systems. This is due to **three major challenges** that such monitoring systems are confronted with.

The first challenge is given by the structure of the data itself: The variance in the measured sensor data records is typically dominated by type and geometry of the workpiece as well as user initiated process adaptations, not by the type of a potential anomaly. This makes generalizing prediction of specific anomaly classes across data sets challenging: Differences in workpiece types and process parameter settings can induce a significant difference in the distributions of recorded data used during training of the models and during prediction, thus the trained model does not match the altered distribution of newly incoming test data anymore. Such differences in training and test data distributions are referred to as *covariate shift* [38] and represent the first major challenge for machine monitoring systems striving for long-term deployment: **Dominance of covariate shift over class-discriminative information in the recorded sensor data.** This dominance makes the standard approach of detecting anomalous signal deviations with a large set of generic features questionable, as scores for these generic features are

typically dominated by the covariate shift in the sensor data distributions and not by the type of a potentially present anomaly. For several types of anomalies however, it is possible to use domain knowledge about physical cause and effect of the anomaly type for the design of specific features custom-built for these anomaly types and thus independent from covariate shift. In part I of this thesis, such features tailor-made for the most important and most common anomaly types of interest are introduced. Classification with these tailor-made features can then typically be performed unsupervised, i.e., without dependency on labels (i.e., annotations of data records).

Such an independence from labels is desirable, as **sparsity of high-quality labels** constitutes the second challenge that automated, model-based machine monitoring systems are confronted with. Most performant machine learning (especially deep learning) models however necessitate large sets of annotated data records. In the second part of this thesis, methods that allow for collecting large sets of labels for the measured sensor data records while introducing a minimum necessary interaction with the annotators and thus minimum additional costs for annotation are proposed. Furthermore, advanced anomaly detection models making optimal use from sparse labels and approaches to estimate the quality of labels before incorporating them into the models are described.

Finally, this thesis aims at performing data evaluation directly on the embedded sensor system (consisting of sensor nodes and gateway system) wherever possible. The main reason for embedded data evaluation is given by an increased data security. Furthermore, reducing the data rate as soon as possible in the time series processing chain reduces the overall system's energy consumption. In addition to constraints imposed by embedded data evaluation, the risk of machine anomalies to cause severe machine damages requires short execution times of trained predictive models in order to allow for fast responses by machine operators (e.g., adjustments of process parameters). The goal of embedded data evaluation and the necessity of timely responses to anomaly reports induce **constraints regarding memory space occupation and model execution time** which constitute the third challenge of this thesis. These constraints are taken into account both in part I of this thesis during the task of proposing computationally simple but effective features custom-made to specific machine monitoring tasks and II of this thesis by designing powerful but scalable neural anomaly detection models.

1.3 Contributions

The two parts of this thesis focus on different challenges of sensor data evaluation as well as various stages of the time series processing chain mentioned in the beginning of this chapter. In Fig. 1.4, these successive stages of the time series processing chain are summarized for the concrete sensor and evaluation scenario considered in this thesis. Here, raw data measured with sensor nodes $S_1, \ldots, S_k$ at k sensor positions are first preprocessed (e.g., filtered or detrended) and segmented. Afterwards, features are extracted from the segmented data. Finally,

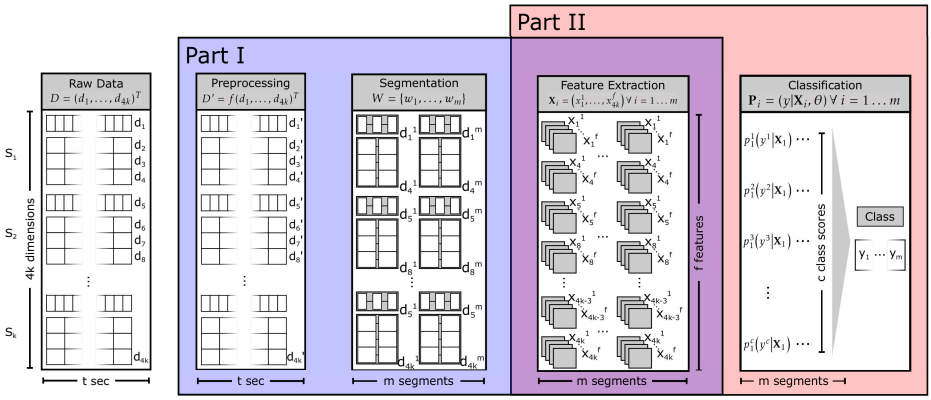


Figure 1.4: *Time series processing chain for evaluation of the sensor data considered in this thesis. Part I of this thesis (highlighted in blue) focuses on simple-to-compute but effective feature extraction by finding a tailor-made segmentation for the specific structure of the sensor data and recovering discrete frequency components related to specific machine parts. Part II (red) focuses on learning features and classifiers directly from the data via generic unsupervised anomaly detection models. In addition, a live annotation approach for low-cost collection of large labeled data sets is introduced and techniques for learning semi-supervised model extensions from sparse and unreliable labels are proposed. Figure inspired from [47].*

classification of time series is performed dependent on these features. As stated earlier, each sensor node consists of a single-axis vibration sensor and a tri-axial acceleration sensor (cf. Table 1.1).

In this thesis, extensions and adaptations of different parts of the time series processing chain are presented in order to match the specific characteristics of the considered data and evaluation systems. The focus of the extensions presented for part I are highlighted in blue in Fig. 1.4, for part II in red (part). In detail, the following alterations of the processing chain are contributed in this thesis.

For the sensor nodes attached to process-related machine parts, data streams illustrate a cyclostationary structure. This is best observable when considering the sensor attached to the workpiece support. The reason for this cyclostationary structure is the same sequence of successive processing steps being applied during the machining of each workpiece. Although expected to occur at deterministic relative locations in the data records due to the fixed machine control program, segment borders do not occur at perfectly similar locations in each data record due to reasons such as package loss during data communication, thus disallowing simple segmentation by hard-coded time instances. In Chapter 3, various existing approaches for signal segmentation are compared and two extensions which incorporate the specific cyclostationary structure of the data in this thesis for a more robust signal segmentation are introduced.

Being able to identify recurrent segments induced by the cyclostationary data structure is beneficial for two reasons: A recurrent segmentation allows to compute features in comparable signal regions in order to construct health indicators for the detection of anomalies with a long-term *drifting* character. Furthermore, detecting deviations from the stationary, recurrent segment structure observed during normal process behavior introduces a computationally efficient approach for detection of *suddenly occurring*, process-related anomalies.

In addition, investigating time-frequency (TF) energy distributions which can be computed from the sensor data proved beneficial for specific prediction tasks. This becomes most obvious for the methods discussed in Chapter 4. There, methods are discussed for recovering discrete frequency components in these time-frequency distributions (TFDs) via estimation of parameters of an imposed signal model per each TFD frame and subsequent tracking of frequency components throughout the temporal course of these frames. Tracking of discrete frequency components and identifying the machine parts they originate from allows for condition monitoring of specific rotating machine parts. This is outlined exemplary for the specific tasks of imbalance detection in rotating machinery (i.e., grinding wheels and dressing wheels) as well as the detection of machined workpieces with an insufficient roundness.

The two chapters that constitute the first part of this thesis are mainly related to comparing and extending various (frequency) preprocessing and segmentation techniques for the sake of extracting tailor-made features for specific machine monitoring tasks. These features allow for a computationally efficient classification for these specific tasks which generalizes across variations of process parameters and workpieces. However, this approach necessitates the predictive tasks to be known in advance. Furthermore, for the design of such tailor-made features, a sufficient amount of data for these tasks has to be accessible in order to analytically study the effect of these anomalies on the data. Often, this is not possible, as potential anomaly types are not known in advance and data is not sufficiently present. On the contrary, detecting deviations from a normal (machine health / production efficiency) state in long-term production floor measurements (i.e., without deliberately provoking anomalies) is often easier to realize. Finally, tracking alterations in the signal segment structure and the location or amplitude of discrete frequency components allows for a computationally efficient detection of certain anomalies. However, anomalies are constrained to manifest in these very phenomena, thus effectively excluding other anomalous phenomena not related to segment structure alterations or drifts of discrete frequency components (e.g., alterations in broader frequency bands or subtle deviations in the signal form not leading to changes in the segment structure).

In part II of this thesis, the focus is thus put on learning time series representations that are independent from covariate shift and can be used for a generic detection of "in-the-wild" recorded anomalies. For this, various anomaly detection models are evaluated on in-the-wild recorded sensor data in Chapters 5 and 6. These algorithms comprise both unsupervised and semi-supervised methods. In order to collect a sufficiently large set for training of semi-supervised anomaly detectors, three approaches are compared.

A typical approach considered in industrial monitoring literature is retrospective annotation of measurement data by domain experts. Anomalies are provoked artificially during measurement campaigns dedicated for specific anomaly types by deliberately altering process parameters. This approach allows to study the influence of certain anomaly types analytically and detached from other phenomena in the data. When domain experts are equipped with additional (e.g., optical) measurements as meta information during annotation, this approach can yield high-quality labels. This is the approach considered during part I of this thesis. However, high quality of annotations is payed dearly for by high annotation costs due to additional time spent by domain experts during retrospective annotation of the sensor data. Furthermore, this approach can only be performed for known types of anomalies and often comes with a risk of causing severe machine damages due to altering process parameters deliberately to insensible values. Finally, the anomalies do not evolve "in the wild". Thus, it is non-trivial to state whether observed signal deviations are restricted to the specific adjusted parameter settings or representative of anomalies as they might appear during everyday machine and process behavior. Ultimately, provoking anomalies always yields a trade-off between predictive quality of anomaly classifiers and annotation costs: While high-performant models rely on huge amounts of labeled data during training, measurement and especially (retrospective) annotation of these data come with non-neglectable costs.

In Chapter 5 of this thesis, a novel live and in situ annotation approach for in-parallel labeling of data records during recording time is suggested as an alternative to the retrospective annotation procedure described above. Sensors are mounted at key points of interest to a grinding machine in a real-world production environment for long-term measurements (several months). Generic anomaly detection algorithms are used to propose suspicious data records for annotation to the end user (i.e., machine operator). As only a small fraction of the data is supposed to be anomalous, depicting only suspicious data records for annotation results in a drastic decrease of annotation effort for the user while offering additional meta information during annotation given by the possibility to inspect both the machine and produced workpieces directly during the annotation process. Both visualization of data records proposed for annotation and the annotation process itself are realized by a self-developed labeling prototype.

The labeling prototype and several design considerations for the prototype are discussed in a user study presented in Chapter 5. In addition, results on the feasibility of the proposed live annotation approach regarding labeling reliability are presented. One of the major findings of this live annotation study is that only anomaly types with a clear, well-known and characteristic signal pattern are identified reliably by the machine operators. Anomaly types manifesting in more subtle and unknown deviating signal patterns however are seemingly harder to identify, resulting in a higher fraction of rejected anomaly propositions and effectively introducing a higher amount of "noise" into the annotation process due to the increased uncertainty of the machine operators. Thus, live annotation comes with low-cost and realistic (i.e., in-the-wild recorded) labels for large sets of data but a higher degree of uncertainty regarding annotation correctness.

In Chapter 6, the focus is on improving quality of annotations by finding more advanced anomaly proposing algorithms than applied in Chapter 5. Various unsupervised neural architectures are discussed and evaluated on the presented data. In addition, a third approach for label collection by automatic generation of probabilistic labels is discussed. Then, novel methods for incorporating the automatically generated, therefore weaker (non-expert) labels into semi-supervised extensions of the unsupervised anomaly detection models are presented. Finally, the benefit of including these labels for creating semi-supervised extensions is compared to both unsupervised models and semi-supervised models trained with expert labels. With these semi-supervised extensions, an end-to-end approach for machine learning based machine monitoring covering all fields of data measurement, data annotation and anomaly detection is created.

1.4 Summary of Contributions and Thesis Outline

The contributions of this thesis can be summarized as follows:

Chapter 3
Different algorithms for segmentation of the cyclostationary structured sensor data evaluated in this thesis are compared. An approach mimicking hidden Markov models (HMMs) by a computationally simpler combination of Gaussian Mixture Models (GMMs) and finite state machines (FSMs) is introduced. Furthermore, an extension to the Bayesian online changepoint detection (BOCPD) algorithm [5] is presented. This extension allows to model the specific, cyclostationary structure in the sensor data. This, in turn, can be used for a more robust segmentation of signals and the successive extraction of health indicators for anomalies with a drifting character as well as the detection of suddenly occurring anomalies in the production process. This will be outlined for two specific types of suddenly occurring anomalies and by the introduction of a novel health indicator for tool condition monitoring.

Chapter 4
Several features custom-built for the specific machine monitoring tasks detection of imbalances in rotating machinery and detection of machined workpieces with an insufficient roundness are presented. For this, methods for recovery of discrete frequency components and successive assignment to machine parts are discussed and evaluated on the sensor data.

Chapter 5
A user study exploring how to collect large sets of labels for rare abnormal events in industrial scenarios and introducing a novel approach for live annotation of sensor streams is presented. To the best of the author's knowledge, no comparable study exists. Other than in the frequent studies on labeling in medical and social applications, labels are not collected via a smartphone-based human–machine interface but via a self-developed visualization and labeling prototype custom-made for harsh industrial environ-

ments. Insights are shared from the process of designing the visualization and labeling interface gathered by exchange with industrial end users (i.e., machine operators). Measures to judge the reliability of reported anomalies and online label feedback in a scenario where neither ground truth labels are accessible nor comparison of labels of multiple annotators is an option are proposed. These assumptions are evaluated on a large corpus (123,942 data records) of in-the-wild recorded industrial sensor data and labels which were collected throughout several weeks. Furthermore, characteristics of anomaly types that can be labeled reliably via live annotation at the proposed visualization and labeling prototype are evaluated.

Chapter 6

A performance comparison of various unsupervised neural anomaly detection models for the detection of anomalies of a grinding machine situated in a real-world factory floor is conducted. Furthermore, a combination of various neural architectures with the Deep SVDD loss function [207] tailormade to anomaly detection is presented. This combination was not discussed in machine monitoring applications before. In addition, a novel weakly supervised anomaly detection loss function building on the Deep SVDD loss function [207] is discussed. This loss function allows incorporating estimates of label uncertainty inherent to the automatically generated probabilistic labels into the process of learning semi-supervised neural anomaly detection models. The loss function was first discussed in a master's thesis [110] supervised by the author of this thesis.

2

Theoretical Background and Related Work

In this chapter, a brief overview of the state of the art research in methods applied throughout this thesis is presented. First, machine health monitoring (MHM) in general and most common MHM tasks as well as related features and models are discussed. A focus is put on the predictive tasks of tool condition monitoring (TCM) and imbalance detection, which are considered among the most important MHM tasks and thus emphasized on in this thesis. Afterwards, different methods from the areas of signal segmentation and estimation of discrete frequencies are discussed, as related preprocessing techniques are extended and applied in Chapters 3 and 4. Sections 2.4 and 2.5 focus on describing the variety of anomaly detection models and human annotation specifics related to the live annotation approach in part II of this thesis. Finally, Section 2.6 describes types of weakly supervised learning and how to improve the quality of weak labels.

2.1 Machine Health Monitoring

The field of sensor-based MHM is inspired by many methodological fields and consequently reached a high and confusing diversity [252]. In order to get an overview of common approaches, the field of methods can be partitioned regarding different criteria. Firstly, one can distinguish between the type of manufacturing process a proposed method can be applied to. An overview of manufacturing processes according to norm DIN 8580 [1] is illustrated in Fig. 2.1. Our focus in this study is mainly in the field of cutting with geometrically undefined cutting edges (e.g., grinding). Secondly, one can group MHM techniques into condition monitoring and process monitoring. While condition monitoring focuses on evaluating the condition of machine parts or the overall machine health state, process monitoring is related to judging quality and efficiency of the machining process. Both fields are in the focus of this thesis. When the goal is not on judging

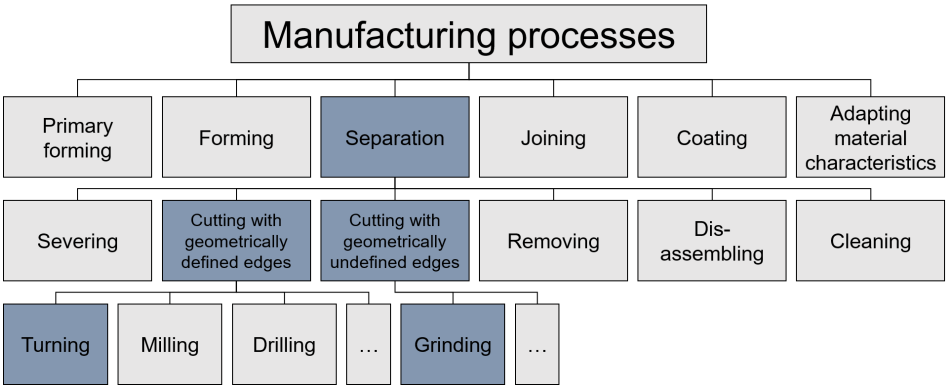


Figure 2.1: *Overview of manufacturing processes (classification according to DIN 8580). The methods applied and proposed in this thesis focus on grinding and turning.*

the current state of the machine, machine part or production process (diagnosis) but predicting some action based on this state (prognosis), both fields are often subsumed under the term predictive maintenance.

Finally, one can divide methods into the stage of the data processing chain they focus on. Jardine et al. propose to distinguish the data processing chain into the three fields of data acquisition, data processing and maintenance decision making [111]. While the first part of this thesis focuses mainly on data processing (i.e., hand-engineering appropriate features for process and condition monitoring), the second part of this thesis focuses on all three of these fields (i.e., how to obtain high-quality measurement data and related annotations, defining advanced models capable of automatically learning appropriate indicators for machine health and process state as well as using them in powerful maintenance decision models).

For data acquisition via condition-based maintenance (CBM) systems, vibration sensors and acoustic emission sensors are the most typical sensor types [111]. Industrially established monitoring systems typically rely on recordings of high-frequency autoencoder (AE) sensors (up to several MHz) [176] and evaluation of the AE root mean square (RMS) signal [175] rather than MEMS sensor evaluation. Regarding the type of data evaluated in CBM systems, the actual sensor data measured via sensors attached to the machines under review are typically in the focus of measurements while context information regarding anomalous events and process adaptations (e.g., minor repairs or machine part changes) is often neglected [111].

Data processing approaches can be further categorized into time domain, frequency domain and TFD techniques. For time domain methods, time synchronous averaging (TSA) and autoregressive models (e.g., autoregressive moving average (ARMA)) are among most widely applied methods [111]. Time synchronous averaging finds the ensemble average of a certain amount of successive raw sensor

signals, attempting to reduce noise, signal fluctuations and other undesired effects, in order to more clearly represent the signal components of interest. The advantage of frequency domain analysis compared to time domain methods is the ability to identify and isolate certain frequency components more clearly than in time domain representations [111]. Frequency domain methods in turn are disadvantageous to TFD methods in that they discard temporal information, which is often relevant in MHM applications, e.g., due to the non-stationary behavior of machine components. Teti et al. complement this generic discrimination into domains of information extraction by a survey on common features applied throughout a diverse set of MHM tasks [252]. A compact overview of the most common feature listed in [252] is given in Table 2.1. As confirmed by the table, most features are general purpose features not tailor-made for specific MHM tasks but relying on statistical measures (e.g., central moments, peak-to-peak values, crest factor and RMS values), time series models (e.g., autoregressive (AR), moving average (MA) and ARMA models), principal component analysis (PCA), singular spectrum analysis (SSA) and frequency domain (FD) transformations (e.g., fast Fourier transform (FFT)) or TFDs (e.g., short-time Fourier transform (STFT), wavelet-based or Hilbert-Huang transforms (HHTs)). These features are opposed to the custom-built features presented in part I of this thesis.

Finally, Tax et al. conducted a study on machine vibration analysis [250] for evaluation of the Support Vector Data Description (SVDD) method introduced by them [249]. As SVDDs are among the most typical outlier detection models until today (cf. recent deep learning publications [207, 208]) and are additionally applied as anomaly detection loss functions in this thesis (cf. Section 6.2), that study is considered relevant to the work presented here. For the evaluation of SVDDs, Tax et al. considered a 64-dimensional vector spanning features based on standard power spectra, envelope spectra, AR models, the MUSIC spectral estimator [219] for estimation of discrete frequencies and classical statistical features (RMS of power spectrum, kurtosis of time domain signal sample distribution and crest factor of this signal). Some of the methods discussed and presented throughout the first part of this thesis are related to similar features, although typically considering slightly different approaches (e.g., estimation of discrete frequencies via more recent non-stationary signal estimators than the outdated MUSIC spectral estimator (cf. Section 2.3)).

Finally, a wide range of maintenance decision making models exists. Among most common choices are HMMs, clustering approaches, support vector machines (SVMs), artificial neural network (ANN)-based approaches, genetic algorithms, Kalman filters and time-dependent proportional hazard models for different probability distributions (e.g., Weibull distributions) [111]. Many of these model types are still being applied in today's MHM literature as confirmed by the more recent survey by Liu et al. [149]. Additional to SVMs and shallow ANN architectures, Liu et al. expanded the list of common MHM classifiers by k-nearest neighbors (kNN) and naive Bayes classifiers. Finally, they outlined the advent of deep learning methods in the field of rotating machinery fault analysis, with an emphasis on AEs as well as deep Boltzman machines and deep belief networks. A

Table 2.1: Overview of most common MHM feature domains

Feature domain	Feature group	Common features
Time domain	General purpose	Statistical, peak-to-peak, crest factor, RMS-related [69, 82, 230]
	Time series models	AR, MA, ARMA coefficients [69, 141, 191, 217]
	PCA-based	2D orbit diagrams [228], feature transforms [2]
	SSA-based	Signal decomposition (trend, detrended signal) [14, 210, 211]
	Entropy-based	Permutation entropy [142]
Frequency domain	FFT-based	RMS- [245], energy- [15] and statistical measures [39] for frequency bands/peaks
Time-frequency domain	STFT-/wavelet-based	RMS- [251], energy- [273] and statistical measures [112, 217] for STFT/wavelet coefficients
	HHT-based	Energy-related [26, 185] measures for Hilbert spectra

recent survey on deep learning in MHM tasks [288] confirmed these deep model types stated in [149], additionally discussing convolutional and recurrent network architectures.

As outlined in this section, the field of MHM methods both regarding data processing (preprocessing, features) and applied models is quite diverse. In the next two sections, a more detailed overview of two specific predictive tasks being in focus of the first part of this thesis is given: Tool condition monitoring (cf. Subsection 3.3.5) and imbalance detection (cf. Subsection 4.3.2).

2.1.1 Tool Condition Monitoring

The goal of TCM is to find an appropriate health indicator reliably measuring the state of the chipping tool. Typically, appropriateness is measured by a smooth and monotonic change of this health indicator across the lifetime of a tool [138]. Based on this health indicator, one can judge the current tool state by classifiers (in the simplest case a fixed threshold) or predict its remaining useful lifetime (RUL).

Additional to these steps of data acquisition, health indicator construction and RUL prediction, Lei et al. discussed dividing the tool lifetime into health stages prior to RUL prediction [138]. Such a subdivision of the lifetime into

health stages is beneficial for a wide range of RUL tasks, e.g., the degradation process of ball bearings [136, 169]. The reason for this subdivision is the creation of separate degradation models incorporating the different degradation statistics in these health stages more precisely. Chipping tools for machine tools like grinding wheels however typically expose a single, continuous health stage as illustrated before for milling tools [10] and presented for grinding tools in Subsection 3.3.5. Thus, a single model is typically sufficient to describe the complete life time of a chipping tool (grinding wheel, milling tool, etc.).

The construction of health indicators can be categorized into physical and virtual health indicators. Physical health indicators are given by measures for physical properties of rotating machinery and most widely applied in the RUL community. Among popular choices are common signal processing measures (RMS values, wavelet and FFT-related coefficients as well as spectral flatness) or statistical time domain measures (e.g., kurtosis of samples, correlation coefficients between successive time series, residual errors and entropy measures) [138]. Virtual health indicators on the other hand are constructed from fusing multiple physical health indicators or multiple signals from different sensors. Common virtual health indicators build mostly on PCA [29, 138], ANNs like self-organizing maps (SOMs) [108, 193], multilayer perceptrons (MLPs) [83] or recurrent neural networks (RNNs) [89] as well as HMMs [174].

Finally, the health indicator can be used either to judge the current tool state (via classification methods) or for RUL prediction. The former and latter differ fundamentally regarding applied methods: While classification for a single health indicator can in the simplest case be performed by a fixed-value threshold, RUL estimation typically involves much more complex models. The latter field of RUL estimators can be subdivided into four groups of approaches: Physical model-based approaches, statistical approaches, data-driven approaches and hybrid approaches [138].

Physical model-based approaches describe the degradation process of the machinery under review by a mathematical model inspired by the underlying physical failure mechanisms. When such failure mechanisms are well understood and can be modeled in detail then physics-based RUL models are highly competitive to other RUL estimators. For many real-world applications however, the underlying mechanical systems are either too complex to be modeled efficiently by a mathematical model (often involving complex particle filter approximations) or not fully understood. This explains the small fraction of physics-inspired models among RUL estimators in the current literature (around 10% according to [138]).

Statistical approaches, on the other side are among most prominent choices for RUL prediction [138]. The most common representatives of this group of statistical approaches are AR methods [192], Markov models [120, 121], proportional Hazard models [61, 131] as well as models based on stochastic processes like the Wiener [229], Gamma [256] or inverse Gaussian process [57, 263]. Among the latter subgroup of stochastic processes, Wiener processes are the most common assumed process behavior as the underlying assumptions match the nature of typical RUL applications (e.g., ball bearing degradation): Wiener process models are represented by a weighted sum of a drift term and a Brownian motion

diffusion term [138].

Instead of building physics-inspired or statistical models describing the degradation process appropriately, one can learn the degradation behavior from given observations with data-driven approaches. Prominent approaches are based on SVMs [50, 70, 235], ANN architectures [108, 154], Gaussian process regression (GPR) models [19, 145] and fuzzy logic methods [55, 262]. Among these methods, GPR approaches are often considered most powerful for RUL estimation due to their high adaptability and the capability to infer sensible and robust degradation models even from a small amount of observations [138].

Finally, hybrid approaches combine methods from all of the above groups attempting to milder the disadvantages of the different groups. Typically, methods from the groups of statistical and data-driven approaches are combined in order to create models which can be learned from existing data but are constrained by an underlying stochastic process model [138].

2.1.2 Imbalance Detection

Although being one of the main reasons for unwanted machine vibrations, the literature on imbalance detection for machine tools is rather sparse. Among published approaches, typical imbalance detection techniques include parameter evaluation for numerical models of rotor and bearing behavior [137, 234], empirical mode decomposition [271, 278] or AR model based features extracted from multi-sensor systems and fused via ANNs [148].

This sparsity in literature is in contrast to the demand in real-world factory floors, where machine tools are typically equipped with industrially established, tailor-made imbalance detection and balancing devices [67]. Such commercially available imbalance-related products are expensive and often involve mechanical modifications at the machine tool in order to place specific sensor nodes at the most convenient places for a high-quality imbalance detection. Thus, the demand in such non-academical industrial environments for a sensor system being able to detect imbalances in combination with other important predictive tasks (like TCM, cf. Subsection 2.1.1) and without the necessity to include any additional sensor devices for this specific task of imbalance detection is high. This necessity is revisited in later sections, where features for imbalance detection are proposed. These features are computed from signals recorded at the generic sensor positions that are used throughout all experiments conducted for this thesis. Thus, the proposed features come without the necessity to include any additional, specifically imbalance-related sensor devices.

2.2 Signal Segmentation

Disclaimer: Parts of this section were taken verbatim from own previous publication [202] ©2019 IEEE.

Popular signal segmentation approaches comprise piecewise linear approximation methods [118], clustering-based methods [143, 212], Hidden Markov

Models [78, 100, 282] and algorithms involving a penalized likelihood function of the data [74, 150, 180].

2.2.1 Piecewise Linear Approximation

Signal segmentation can be approached by segment-wise linear approximation of the signals, i.e., segment borders are found at end points of linearized segments. Piecewise linear segmentation was originally approached either by sliding window approaches, bottom-up or top-down approximation of signal segments [118]. In sliding window approaches, segments are extended starting from a starting raw data point (*anchor point*) until some threshold on an approximation cost measure (e.g., Euclidean distance) is exceeded. The previous segment is then defined to have ended at the previous raw data point and a new segment ist started from the current raw data point. While this simple technique has the advantage of being online-operable (i.e., working on streaming data) it can not produce a globally optimal solution as only points in a local neighborhood following the anchor point can be considered for segmentation [33].

Bottom-up approaches address this disadvantage of sliding window approaches by iteratively merging cheapest pairs of segments until a cost threshold is met. These approaches thus take the whole data set into account and can often be found to be more accurate than sliding window approaches [118]. However, due to the necessity of the whole data being present for iterative merging of segment pairs, the segmentation can no longer be performed online. Also, the complexity increases from $\mathcal{O}(n)$ to $\mathcal{O}(n^2)$ (with n being the length of the signal considered for segmentation).

In [118] a combination of bottom-up approximation of a buffer of raw data samples and a sliding window step for adding raw data points to the buffer was proposed. Due to only considering raw data samples buffered in the sliding window, an online-capable bottom-up approach emerged. Consequently, due to the combination of sliding window and bottom-up methods, the method was termed SWAB.

The applicability of piecewise linear approximation (PLA) approaches relies heavily on the nature of the data: Piecewise linear approximations are justified for signals with a rather low-varying information content in the segments. This is not the case for the raw sensor data in this thesis. Thus, PLA approaches are not further considered for signal segmentation.

2.2.2 Clustering-based Approaches

Signal segments can be identified by clustering approaches after transformation of successive signal subsequences into a potentially high-dimensional feature space. First, each signal is divided into fixed-length blocks of a prespecified length and features are extracted for each of these blocks. Then, for each feature vector extracted from a single block the most probable membership to any of the clusters is estimated, thus ending up with a vector of most probable cluster

memberships for successive feature vectors extracted from each signal. Finally, segment borders are assigned at transitions between cluster memberships.

Simple parametric models like GMMs can be powerful methods for identification of coherent clusters in the feature space. However, they lack in modeling the temporal latent behavior of successive signal samples: While GMMs treat successive feature vectors as independent and identically distributed (iid), they are actually temporally correlated for the given data. Thus, cluster membership is highly probable to stay similar among multiple successive feature vectors. Clustering approaches that model correlations between successive feature scores extracted from the signals lend themselves naturally for the given time series segmentation problem, as features allow to flexibly capture the underlying statistics of the data-generating process and its changes across time. One of the most prominent representatives of this class of time-dependent clustering methods is the HMM.

Signal Segmentation based on Hidden Markov Models

HMMs are widely used models for time series classification and come with inherent segmentation of the signals [40]. They are suited for a wide range of temporally structured prediction tasks by allowing to incorporate prior knowledge into the learning process.

Formally, any HMM model $\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$ can be defined by the following parameters [232]:

- An $S \times S$ transition matrix $\mathbf{A}$, where S specifies the number of states. The matrix entry A_{ij} specifies the probability for a transition from state i to state j. If the transition probabilities a_{ij} are constant, i.e., independent of time t, the Markov process is a stationary one.

- An observation model $\mathbf{B}$ for each of the S states $s_1, \ldots, s_S$. When assuming normally distributed observations, each observation model $\mathbf{B}$ is parameterized by multivariate normal distributions $\{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$. For a dimension M of the features space $\boldsymbol{\mu}$ is an $M \times 1$ mean vector and $\boldsymbol{\Sigma}$ an $M \times M$ covariance matrix.

- An $S \times 1$ probability vector $\boldsymbol{\pi}$, which defines prior probabilities of the HMM's initial states.

The parameters $\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$ of the HMM model are typically learned via maximum likelihood methods. This is specified in the next paragraph. The parameters are learned relying on the following quantities for N training data records:

- An $M \times T$ observation matrix $\mathbf{O}_i$ per each of N training data records. Each $\mathbf{O}_i$ consists of a sequence of T entries $\mathbf{o}_1, \ldots, \mathbf{o}_T$. Each observation vector $\mathbf{o}_t$ is represented by an M-dimensional feature vector.

- A $T \times 1$ vector of hidden HMM states $\mathbf{z}_i = z_1, \ldots, z_T$ for all training observation matrices $\mathbf{O}_1, \ldots, \mathbf{O}_N$.

Learning of Model Parameters Learning the parameters $\{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$ of the HMM model being given an observation sequence $\mathbf{O}$ is classically performed via the Baum-Welch algorithm [195]. The objective being optimized for maximum likelihood parameter learning is:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} p\Big(\mathbf{O}|\boldsymbol{\theta}\Big) \tag{2.1}$$

As the observations are not considered independent and identically distributed (iid), the likelihood objective does not factorize into $p(\mathbf{O}|\lambda) = \prod_{t=1}^{T} p(\mathbf{o}_t|\boldsymbol{\theta})$. Instead, the Baum-Welch algorithm makes use of expectation-maximization (EM) techniques, an iterative method to find either maximum likelihood or maximum a posteriori (MAP) parameter estimates $\boldsymbol{\theta}^* = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$. When assuming normal distributed observation models $\mathbf{B}$ as specified above, $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ have to be learned for each of the j hidden states. The Baum-Welch algorithm consists of two alternating steps and is described here for maximum likelihood parameter estimation [168]:

1. E step: The expectation (E) step consists of computing a function $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n-1)})$ which represents the expectation value $\mathbb{E}$ of the log likelihoods $L(\boldsymbol{\theta}; \mathbf{O}_i, \mathbf{z}_i) = p(\mathbf{O}_i, \mathbf{z}_i|\boldsymbol{\theta})$ for the $i = 1, \ldots, N$ training observations $\mathbf{O}_i$ and hidden states $\mathbf{z}_i$ based on the current parameter estimates $\boldsymbol{\theta}^{(n-1)}$:

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n-1)}) \triangleq \mathbb{E}\Big[\log\, p(\mathbf{O}_i, \mathbf{z}_i|\theta)\Big] \tag{2.2}$$

2. M step: The maximization (M) step aims at finding an approximate estimate of the maximum of the log marginal likelihood $L(\boldsymbol{\theta}; \mathbf{O}) = p(\mathbf{O}|\boldsymbol{\theta}) = \int p(\mathbf{O}, \mathbf{z}|\boldsymbol{\theta}) d\mathbf{z}$ utilizing the function $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n-1)})$ found during the E step:

$$\boldsymbol{\theta}^{(n)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n-1)}) \tag{2.3}$$

Closed form update rules for parameters $\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$ can be specified as follows [40]:

$$\boldsymbol{\pi}_j^{(n)} = \frac{1}{N} \sum_{i=1}^{N} p(z_{i,1} = j|\mathbf{O}_i; \boldsymbol{\theta}^{(n-1)}) \tag{2.4}$$

$$A_{jk}^{(n+1)} = \frac{\sum_{i=1}^{N} \sum_{t=2}^{T} p(z_{i,t-1} = j, z_{i,t} = k)|\mathbf{O}_i; \boldsymbol{\theta}^{(n-1)}}{\sum_{i=1}^{N} \sum_{t=2}^{T} p(z_{i,t-1} = j|\mathbf{O}_i; \boldsymbol{\theta}^{(n-1)}} \tag{2.5}$$

$$B_j^{(n+1)}(k) = \frac{\sum_{i=1}^{N} \sum_{t=1}^{T} p(z_{i,t} = j)|\mathbf{O}_i; \boldsymbol{\theta}^{(n-1)} \mathbb{I}(\mathbf{o}_{i,t} = k)}{\sum_{i=1}^{N} \sum_{t=1}^{T} p(z_{i,t} = j)|\mathbf{O}_i; \boldsymbol{\theta}^{(n-1)}} \tag{2.6}$$

where $z_{i,t}$ refers to the t-th element of the i-th hidden state vector $\mathbf{z}_i$ and $\mathbb{I}(c)$ to an indicator function with elements being 1 if the condition c holds and 0 otherwise.

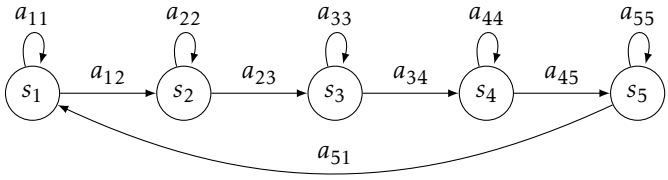


Figure 2.2: *5-state-example of a circular L2R HMM architecture. The L2R architecture allows modeling a recurrent structure in the data.*

After convergence, the final parameter set can be used to estimate the cluster membership of new data points. These membership estimates are given as posterior probabilities and the amount of similarity between the probability estimates of different classes can be interpreted as an uncertainty measure to design a soft threshold for assigning class labels. Compared to k-means, the additional covariance terms allow learning clusters of different expansion.

By initializing elements of the state transition matrix $\mathbf{A}$ to zero one can enforce constraints on state transitions to follow a strict temporal order. This allows, for example, learning a left-to-right (L2R) structure as depicted in Fig. 2.2. By allowing a transition from the last state to the first a circular L2R structure can be obtained.

During parameter learning, the number of states S can be considered constant. In general however, this is a further model hyperparameter which needs to be optimized. Finding this optimal number S of states to explain the data is typically referred to as model selection.

Model Selection Model selection for HMMs is typically performed via information-theoretical criteria (Akaike information criterion (AIC), Bayesian information criterion (BIC) and others) or likelihood ratio tests [60]. All of these methods impose iid generated data by performing model selection via maximizing a regularized version of the likelihood $p(\mathbf{O}|\boldsymbol{\theta})$.

Thus, the temporal correlation of the successive signal samples is not taken into account. This can be disadvantageous for a scenario like the one summarized in Fig. 2.3: When neglecting the temporal dependency of successive observations, a 3-state HMM is identified as the optimal model, as feature scores of the observations overlap for cluster S_3. Consequently, the model can not learn the correct state transition possibilities as it would be possible when considering the temporal structure of the data by the 4-state HMM (Fig. 2.3, right). This scenario is close to the signal segmentation problem for the data considered in this thesis and illustrates the benefit of incorporating the temporal structure of the data into model selection.

Siddiqi et al. proposed STACS (Simultaneous Temporal and Contextual Splitting), an efficient top-down algorithm for HMM model selection by repeated state-splitting [231]. Beginning from an prespecified number S_0 of initial states,

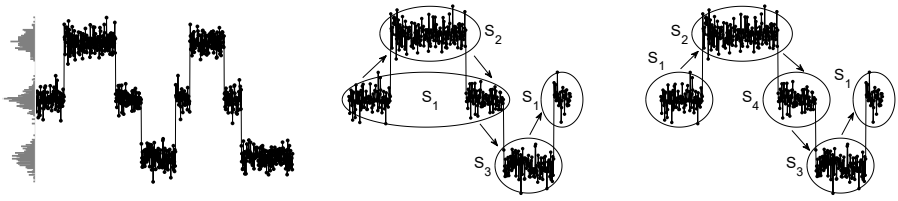


Figure 2.3: *Left: Sequence of feature scores and histogram of feature score occurrences. Middle: Models only considering contextual properties fail to identify the correct state transition behavior (by modeling similar feature scores as one state) for sequential data. Right: STACS considers both contextual and temporal properties during model selection and identifies a sensible set of states. Figure inspired from [231].*

during each state-splitting step i potentially better candidate models are created by splitting either of the $S_i = S_0 + i$ states. These candidate models $\boldsymbol{\theta}_i = \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{S_i}$ are created by constraining $\boldsymbol{\theta}_i$ to the model $\lambda = \boldsymbol{\theta}_{\boldsymbol{i}}^*$ from the former state-splitting step except the observation model parameters $\mathbf{B}_i$ of offspring state s_i and all its input and output transition probabilities in $\mathbf{A}$. The candidate models $\boldsymbol{\theta}_i$ are compared to each other regarding the partially observed Viterbi path likelihoods $p(\mathbf{O}, \mathbf{z}^*_{\setminus \mathcal{D}_i} | \boldsymbol{\theta}_i)$. Here, $\mathcal{D}_i$ denotes the subset of observations $\mathbf{o}_t$ owned by state s_i in the Viterbi path and $\setminus \mathcal{D}_i$ specifies that all observations are fixed at their optimal Viterbi path states $\mathbf{z}^*$ during this parameter estimation step except the ones in $\mathcal{D}_i$ [231].

From models $\boldsymbol{\theta}_i$ scored regarding their partially observed Viterbi path likelihoods, the best-scoring model is compared to the un-split model λ based on the information criterion BIC. Finally, λ is updated with the best-scoring model $\boldsymbol{\theta}_i$ if the latter's BIC is lower than the one of λ. Otherwise, model selection is terminated and the current λ is returned as the final, optimal model.

2.2.3 Changepoint Approaches

When the data-generating process is assumed to be piecewise stationary, changepoint methods represent an additional family of approaches to divide signals into segments resembling this piecewise stationarity. Changepoints are defined as point in the signal where abrupt changes in the parameters of this data-generating process occur. In the following sections, different strategies for changepoint detection are summarized and well-known representatives of these different algorithmic families are named. This work focuses on unsupervised, online computable segmentation models. A broad overview of changepoint detection methods additionally covering subspace models [42, 116, 277], kernel-based [93] and graph-based [56] methods can be found in [16].

Methods based on Likelihood Ratios of the Data

A large fraction of changepoint methods is based on either direct or indirect estimation of ratios of likelihood $p(\mathbf{x})$ and $p'(\mathbf{x})$ of the data $\mathbf{x}$ in adjacent signal subsequences. A detailed overview of different algorithms focusing on these strategies can be found in [150]. Among most prominent representatives of indirect density ratio estimators are CUSUM [180], GLR [268] and Change Finder [246], most popular direct density ratio estimators are KLIEP [244], uLSIF [115] and RuLSIF [276]. Although in general appealing and successful in a wide range of time series segmentation applications, neither of these likelihood ratio based methods proved suitable for the data evaluated in this thesis. Thus, as these methods are not further considered in this thesis, the author refrains from explaining the methods in detail.

Bayesian Online Changepoint Detection (BOCPD)

Similar to likelihood ratio based methods in the previous paragraph, the BOCPD algorithm introduced in [5] allows dividing signals into non-overlapping segments of stationary generative data distributions between changepoints. Other than the likelihood ratio based methods, BOCPD relies on methods from the field of Bayesian statistics for modeling of the probability distributions in consecutive data segments and thus in finding changepoints. Different work extending BOCPD to model data-generating distributions more flexibly [209, 253] or to use changepoint information for the sake of robust time series predictions [80] emerged quickly.

Assume the goal is predicting future samples x_{t+1} depending on previously observed samples $\mathbf{x}_{1:t}$ from a sensor stream up to the current time step t. In a Bayesian context, this involves a predictive distribution $p(x_{t+1}|\mathbf{x}_{1:t})$. When the sensor stream exhibits a piecewise stationary sample-generating process, then the predictive distribution should rely on the samples from the current stationary segment only. In a Bayesian context, this can be done elegantly by conditioning the predictive distribution on a latent variable. BOCPD seizes this idea by introducing a latent *run length* variable r_t [5], which is defined as the distance to the last changepoint having occurred in the data. Then, the predictive distribution $p(x_{t+1}|\mathbf{x}_{1:t})$ is obtained by integrating over the posterior distribution $p(r_t|\mathbf{x}_{1:t})$ on the current run length r_t:

$$p\big(x_{t+1}|\mathbf{x}_{1:t}\big) = \sum_{r_t} p\big(x_{t+1}|r_t, \mathbf{x}_t^{(r)}\big) p\big(r_t|\mathbf{x}_{1:t}\big) \tag{2.7}$$

Here, $\mathbf{x}_t^{(r)}$ are observations associated with the current run r_t, i.e., the last r_t observations of $\mathbf{x}_{1:t}$ [209]. When the focus of interest is on finding the most probable estimate of the current run length r_t, this can be done efficiently by finding the

maximum of the conditional posterior distribution

$$p\left(r_t|\mathbf{x}_{1:t}\right) = \frac{p\left(r_t, \mathbf{x}_{1:t}\right)}{p\left(\mathbf{x}_{1:t}\right)}. \tag{2.8}$$

Henceforth, this conditional posterior distribution is referred to as *run length distribution*. As probability mass of the run length distribution is highly concentrated at a few peaks, pruning of run lengths with a probability below a threshold (e.g., $\epsilon = 10^{-4}$) can be applied. For equally-spaced changepoints, this reduces run time from $\mathcal{O}(T^2)$ to $\mathcal{O}(T)$ as outlined in [209, 253]. The approach was initially suggested in [5].

The distribution $p(r_t, \mathbf{x}_{1:t})$ can be found recursively [5]:

$$p\left(r_t, \mathbf{x}_{1:t}\right) = \sum_{r_{t-1}} p\left(r_t|r_{t-1}\right) p\left(x_t|r_{t-1}, \mathbf{x}_t^{(r)}\right) p\left(r_{t-1}, \mathbf{x}_{1:t-1}\right) \tag{2.9}$$

The right-hand side of Eq. 2.9 consists of three terms:

1. The predictive distribution $p(x_t|r_{t-1}, \mathbf{x}_{1:t})$ collapses to $p(x_t|r_{t-1}, \mathbf{x}_t^{(r)})$, thus depending only on recent $\mathbf{x}_\mathbf{t}^{(\mathbf{r})}$.

2. A joint distribution $p(r_{t-1}, \mathbf{x}_{1:t-1})$ from time step $t-1$.

3. A conditional prior distribution $p(r_t|r_{t-1})$ on changepoints (i.e., $r_t = 0$). Adams et al. proposed to define it as follows for efficient computation (nonzero probability mass only for outcomes $r_t = 0$ and $r_t = r_{t-1} + 1$) [5]:

$$p\left(r_t|r_{t-1}\right) = \begin{cases} H\left(r_{t-1}+1\right) & \text{if } r_t = 0 \\ 1 - H\left(r_{t-1}+1\right) & \text{if } r_t = r_{t-1}+1 \\ 0 & \text{otherwise} \end{cases}$$

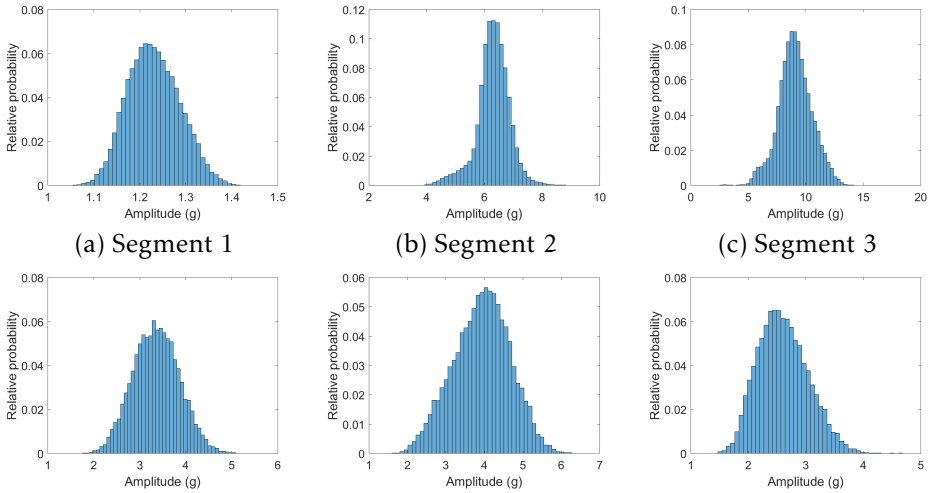
The function $H(\tau)$ is named *hazard function* [75]. In the simplest case, an uninformative constant hazard function $H(\tau) = 1/\lambda$ can be chosen as discussed in [5]. This results in making changepoint estimates $p(r_t = 0|r_{t-1})$ independent of r_{t-1}. Here, λ is a constant timescale parameter which has to be defined in advance or can be treated as a further model hyperparameter which has to be optimized [209, 253].

For the experiments presented in this thesis, iid normal observations x_t and a Normal-Inverse-Gamma parameter prior $p(\mu, \sigma^2|\mu_0, \kappa, \alpha, \beta)$ are assumed in accordance with [5, 209]:

$$x_t \sim \mathcal{N}\left(\mu, \sigma^2\right) \tag{2.10}$$

$$\mu \sim \mathcal{N}\left(\mu_0, \sigma^2/\kappa\right) \tag{2.11}$$

$$\sigma^{-2} \sim \text{Gamma}\left(\alpha, \beta\right) \tag{2.12}$$



(a) Segment 1 (b) Segment 2 (c) Segment 3

(d) Segment 4 (e) Segment 5 (f) Segment 6

Figure 2.4: *Distribution of signal samples x_t across all segments 1 to 6 of an exemplary signal of the measurement data considered in this study. All segments illustrate stationary unimodal distributions which can be reasonably well approximated by univariate normal distributions for computational convenience.*

Here, α and β are the shape parameter and rate parameter of the Gamma distribution and κ acts as a scaling factor for the variance σ^2. The choice of a univariate normal distribution as prior distribution for the data x_t is sensible for the sensor data considered in this thesis as verified in Fig. 2.4: The data illustrate piecewise stationary and unimodal distributions across 6 signal segments which can be reasonably well approximated by a univariate normal distribution. Samples of an exemplary envelope signal are illustrated, as this is the representation of sensor signals which is used for segmentation in later chapters.

In addition, these choices of data and parameter priors are computationally convenient: As prior $p(\mu, \sigma^2|\mu_0, \kappa, \alpha, \beta)$ and posterior $p(\mu, \sigma^2|\mathbf{x}_{1:t})$ form a conjugate pair for the assumptions made above, updates of parameters $\{\mu_0, \kappa, \alpha, \beta\}$ yield a closed form solution [167].

BOCPD Extensions Relying on Further Assumptions of the Data Structure

Different approaches to robustify BOCPD changepoint estimation have been proposed. In addition to theoretically well-founded but computationally expensive generalized bayesian inference (GBI) methods as described in [126], approaches incorporating assumptions about the data structure into the changepoint estimation process were introduced, which typically come with a lower computational effort than GBI methods.

In [269], Wilson et al. proposed a hierarchical extension of the BOCPD ap-

proach. Their approach allows inferring both (globally) constant or (locally) adaptive estimates of the typical frequency (hazard rate) of changepoints and thus allows creating a more informative run length prior distribution $p(r_t|r_{t-1})$ than the one in [5]. However, it does not allow to model a recurrent behavior of run length patterns, i.e., a vector of run lengths r_t that reoccurs for each recorded sensor signal.

Maslov et al. proposed an approach explicitly dedicated to modeling recurrence of changepoints in [159]. Here, recurrence was defined by quasi-periodicity, i.e., by assuming periodic recurrence of changepoints while allowing small deviations of individual changepoints from this periodic behavior. For this purpose, a predictive change confidence function (PCCF) was embedded into the Bayesian changepoint detector. The PCCF was used as a more informative hazard rate h to be included into the run length prior $p(r_t|r_{t-1})$ than the constant hazard rate proposed in [5]. However, the approach does not allow to model a generic recurrent (but non-periodic) structure of data like the changepoint recurrence distribution (CPRD) estimator introduced in later sections does.

2.3 Modeling Non-Stationary Frequency Components

As outlined in Section 2.1, machine health monitoring (MHM) features for machine-part-related condition monitoring can be related to the detection of stationary or tracking of non-stationary discrete frequency components. The main steps of this approach are estimation of parameters of an imposed signal model and subsequent connection of spectral peaks based on these parameter estimates in order to obtain frequency component tracks.

2.3.1 Parameter Estimation

Superpositional sinusoidal signal models are the most common signal model assumptions imposed in the generic field of line spectrum estimation [239]. Methods like the distribution derivative method (DDM) [36] allow estimating the complex parameters $\boldsymbol{\alpha}$ of this signal model and thus specifying amplitude, frequency and phase of the generalized sinusoidal frequency components in the signal model in a convenient way (cf. Subsection 4.2.1 for details). Amplitude, frequency and phase estimates in turn prove useful in the following step of frequency component tracking, where the frame-wise signal estimates $s(t)$ have to be connected in a sensible way to identify the desired non-stationary evolving frequency components.

Sinusoidal parameter estimation is a generic problem arising in multiple application domains involving radar [51], wireless communications [25] or direction of arrival estimation for sensor arrays [156, 178]. The goal of estimating parameters of a signal model of superposed (complex) sinusoids is referred to by the term of line spectrum estimation. When assuming stationary behavior, the line spectrum estimation problem and respective signal model can be defined as follows [92]:

Definition 2.1 (Stationary line spectrum problem) For a stationary signal model

$$x(t) = \sum_{j=1}^{R_k} \psi(\theta_j)\alpha_j, \tag{2.13}$$

estimate the model order R_k alongside the normalized frequencies $\theta_j \in [0, 1]$ and complex coefficients $\alpha_j \in \mathbb{C}$ for each signal component $j = 1, \ldots, R_k$.

Here, $\psi(\theta_j) : [0, 1) \to \mathbb{C}^{N\times 1}$ is a vector of Fourier components with nth entry $[\psi(\theta_j)]_n \triangleq \exp(i2\pi(n-1)\theta_j)$ indexed by the frequencies θ_j for $n = 1, \ldots, N$.

Early approaches for solving this stationary line spectrum estimation problem were suggested by subspace methods, among which multiple signal classification (MUSIC) [219] and the Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) approach [206] are prominent representatives. Both methods estimate the line spectrum parameters based on covariance characteristics of signals and successive eigenvalue decomposition. They rely on prior knowledge of the model order or estimating it in a separate estimation step (i.e., additionally to estimating frequency components and mixture parameters). When the model order is mis-specified, these methods illustrate large decreases in estimation performance.

Addressing this problem of model order mis-specification, grid-based sparse estimation approaches were introduced with [156], [107] and SPICE [240] among their most prominent representatives. Grid-based sparse estimators are inspired by compressive sensing (CS) techniques and simplify former line spectrum estimation techniques by inducing a sparse reconstruction problem where frequencies are constrained to a finite grid. These methods are advantageous compared to subspace methods in that they implicitly estimate the model order R_k. However, they come with an inherent trade-off between accuracy of frequency estimation and computational complexity, governed by the grid resolution (i.e., spectral resolution defined by the Rayleigh limit $\frac{1}{N}$) [92]. Consequently, off-the-grid extensions like GLS [280], AST [37] and its extension WANM [281] were suggested. These come with the same advantages of grid-based methods (inherent model order estimation, sparsity-inducing) but are not limited to a fixed grid. Unfortunately, these off-the-grid extensions scale at least cubically in size N of the spectral elements [92].

Complementary to the latest directions of line spectrum estimation being formulated as finite sparse reconstruction problems, Bayesian estimation approaches entered the field of line spectrum estimation. Compared to (stochastic) maximum likelihood approaches as in [178], they allow modeling assumptions about the structure of the line spectrum estimation approaches in prior distributions on the signal model coefficients. Zachariah et al. [283] built upon this possibility of including prior assumptions into a probabilistic formulation very similar to the model suggested in 2.13. They related their work to the subspace-based approach in [270], but mentioned that this approach implies deterministic knowledge of a subset of frequencies while assuming no prior knowledge about the remaining frequencies. They explicitly referred to the sparse amount of literature on proba-

bilistic treatment of the line spectrum estimation problem [45, 71]. The work of Zachariah et al. in [283] inspired a multitude of extensions like variational line spectral estimation (VALSE) [22] and its extensions MVALSE [293] and VALSE-EP [294]). Many more recent approaches model these assumptions by sparsity-inducing priors (a list of references can be found in [92]). Such sparsity-inducing priors yield inherent model order estimation opposed to the stochastic maximum likelihood approaches and early subspace approaches but illustrate a high computational complexity (cubic in the number of signal model components).

Finally, in [92], an approach combining the ideas of Bayesian approaches and off-the-grid methods promising computational complexity of $\mathcal{O}(N \log^2 N)$ was introduced.

The above approaches typically built on the signal model 2.13 that assumes stationary behavior of frequency components. In [36], Betser et al. discussed a generalized sinusoidal signal model similar to Eq. 2.13 capable of capturing non-stationary behavior both in location and amplitude of these frequency components:

Definition 2.2 (Non-stationary line spectrum estimation problem) For a non-stationary signal model

$$x(t) = \sum_{j=1}^{R_k} \exp\left(\sum_{i=0}^{Q} \alpha_{ij} t^i \right), \tag{2.14}$$

estimate the model order R_k alongside the complex coefficients $\alpha_{ij} \in \mathbb{C}$ for each signal component $j = 1, \ldots, R_k$.

Thus, the Fourier vector $\psi(\theta_j) : [0, 1) \to \mathbb{C}^{N \times 1}$ from the stationary line spectrum estimation problem in Def. 2.1 is replaced by a sum of monomials t^i of order Q weighted by the complex coefficients α_{ij}. The weighted sum of monomials allows to approximate frequency components that generalize sinusoidal components. The terms $\sum_{i=0}^{Q} \alpha_{ij} t^i$ are thus referred to as *generalized sinusoids*.

The methods for estimation of non-stationary sinusoidal parameters introduced in [36] built on constructing a linear equation system via DDM. This DDM-based approach has become state of the art for non-stationary sinusoidal parameter estimation in frequency component tracking applications [171, 243] and are also used in the conducted experiments to model the non-stationary behavior of discrete frequency components.

2.3.2 Frequency Component Tracking

When aiming to track non-stationary frequency components, TF peaks detected in each TFD frame have to be connected in a sensible way. In the following section, an overview over methods for tracking of frequency components leveraging the information obtained during parameter estimation as presented in the previous subsection is given.

Early literature on tracking of time-evolving, discrete frequency components

focused mainly on linear prediction techniques [133] or probabilistic state space models like Kalman filters [162, 214] and HMMs [64]. The seminal works of McAulay et al. [160] and Depalle et al. [64] introduced approaches for frequency component tracking in sinusoidal signal models via matching of successive spectral peaks based on their related parameter estimates. Both methods estimate peaks and related parameters from STFT spectrograms. For matching of successive spectral peaks, they impose constraints on the continuity of parameter slopes. McAulay et al. used a greedy algorithm to identify peaks and connected them based on a matching criterion favoring small differences of frequency estimates of successive peaks [160]. Depalle et al. on the other hand formulated the matching criterion as a HMM (trajectory) optimization problem and relied both on amplitude and frequency estimates [64].

Kereliuk et al. extended the work of Depalle et al. by augmenting the HMM matching criterion with explicit chirp rate estimates [119]. In order to find these chirp rate estimates, Wigner-Ville transform (WVT) and Hough transform (HT) were combined for parameter estimation. Making the matching criterion dependent on chirp rate estimates improved the tracking quality compared to [64] for different scenarios, e.g., in order to resolve close-by frequency component tracks [119].

In [243], Stowell et al. extended the previous methods in [64, 119, 160] by relying on the DDM for parameter estimation as initially described in [36]. Application of the DDM method improves over simple spectrogram representations or generalized reassigned spectrogram representations [267] and can use any linear transform (e.g., Fourier transform or Wavelet transform) or combinations of them [36]. In addition, Stowell et al. relied on using a Markov renewal processs (MRPs) for matching of detected peaks via related parameter estimates. Compared to HMMs, MRP come with a natural way to allow discontinuities of frequency components (so-called ”sleep states”) during matching of successive spectral peaks. Being able to model such discontinuities is valuable when signals expose a high noise level and consequently result in frequent spurious spectral peaks (cf. Subsection 4.3.1). Such spurious peaks lead to a decrease in concentration of TFD energy along the wanted frequency components and thus many, small-length and randomly directed frequency component tracks in the TF plane.

The above methods building on Markov state estimation problems (HMMs, MRPs) can quickly become computationally intractable as the number of states grows exponentially with the number of peaks detected per analysis frame [119] leading to an overall factorial complexity [73]. For noisy signals as in the considered measurement data, spurious detected peaks lead to a high amount of detected peaks. Neri et al. proposed an alternative to these robust Markov-based estimators by framing the task of matching successive spectral peaks via a linear optimization problem [171]. Solving this linear optimization problem is faster than the Markov methods described above (only polynomial complexity in overall [73]) and is thus applied in this work as method of choice due to the computational constraints mentioned in Chapter 1. Similar complexity is reached with the linear programming methods applied for frequency component tracking in [73].

The above methods for frequency component tracking were mainly proposed

for application in the audio domain [64, 73, 119, 171, 243, 267]. For industrial applications, the majority of literature on frequency component tracking focuses on order tracking, where integer multiples (orders) of a fundamental frequency instead of absolute frequencies (Hz) are used as basic atoms of frequency analysis techniques [79]. Among the most popular applications for order tracking are monitoring of rotating machinery [79, 289] and monitoring of gearboxes, e.g. in wind turbines [18, 81] or helicopters [179].

There are three major fields of methods applied for order tracking [43]: Methods based on computed order tracking [79], on Vold-Kalman filtering [260] and integral transforms (typically the Fourier transform) that directly estimate the order domain from the time domain signal [41]. The most similar application to the MHM scenario in this thesis is given in [81]. Authors aimed to perform condition monitoring for a wind turbine by extracting spectral peaks via the approach of McAulay et al. [160], which were subsequently connected to create frequency component tracks. Spectral peak extraction as proposed in [81], however, comes without estimation of complex parameters α like in [36] and thus misses important information concerning amplitude of these peaks in the matching criterion for frequency component tracking. Furthermore, the approach in [160] assumes stationary frequency component behavior. Finally, authors constrain frequency components to form a set of harmonics, which is less general than the approach in [36].

2.4 Anomaly Detection

Disclaimer: Parts of this section were taken verbatim from own previous publication [201].

Different types of anomalies can be distinguished regarding their relation to the rest of the data. In this thesis, the focus is on *collective anomalies* [53]. This type of anomalies is characterized by a collection of signal samples being interpreted as anomalous behavior and opposed to *point anomalies*, which manifest in single outlying signal samples. Furthermore, anomalies considered in this study manifest as *contextual anomalies*, where the context of signal samples (e.g., relative position in the signal) is relevant for an outlying segment of data being labeled anomalous.

For this intersection of collective and contextual anomalies, a large corpus of potential anomaly detection models can be considered. These models can be distinguished based on the representation of the data used as input for the model:

- One-dimensional representation: Anomaly detection models rely on the data being given as one-dimensional vectors. These vectors can be given either as raw signals or a transformation of the data to another, one-dimensional representation. Popular transformations are envelope signals [35], wavelet-based representations [227] or other spectral transformations based on singular value decompositions [144].
- Multi-dimensional representations: These representations emerge when the sensor data are projected to a dual space by extraction of features. When

aiming for a generic anomaly detection model, the major challenge is given by the choice of a generic but expressive set of features [181]. Among popular choices are statistical measures and wavelet-based features [252] or filter bank features (e.g., Mel-frequency cepstral coefficient (MFCC) features) [30]. The latter yield similar information to anomaly detection approaches based on TFDs.

- TFD representations: This group of two-dimensional representations can be interpreted as a subgroup of the former list item of multi-dimensional representations. Recently, different powerful deep learning approaches capable of learning the latent representations of the underlying, data-generating process have been introduced (with a focus on two-dimensional representations, typically images). Among these, deep generative models like variational autoencoders (VAEs) [124], generative adversial networks (GANs) [86], auto-regressive generative models like PixelRNN/CNN [177] and non-autoregressive flow-based models [65, 66, 123] supersede earlier AE approaches [204, 258, 298] which come with a compressed latent representation of the data but without the possibility of generating samples from the latent representation. It is this ability to sample from the generative process of the data which seems to allow deep generative models to capture details of the data flexibly without any access to labels.

2.4.1 Shallow Models

Methods based on one-dimensional representations Approaches of direct clustering and classification of one-dimensional time series representations rely on the computation of pairwise time series distance measures. The most common measures are euclidean distance (ED) and dynamic time warping (DTW) distance [34] as well as its extensions (soft-DTW (SDTW) [63], DTW barycenter averaging (DBA) [186], etc.). While Euclidean distances are calculated directly based on the samples at corresponding signal locations, DTW-related measures come with an additional, preceding step for optimal alignment of signals via nonlinear warping of the time series. This flexibility allows comparison of signals with different lengths or non-uniformly, affine transformed signals. Unfortunately, DTW measures involve solving the optimization problem for signal alignment with dynamic programming techniques, thus trading off their flexibility with an increased computational complexity compared to simple Euclidean distance measures: DTW scales quadratically with the length n of the time series. In order to reduce this computational cost, several approximation techniques were proposed. Among these, LB_Keogh is one of the most popular [117]. LB_Keogh scales linearly with the length n.

For classification, kNN and especially 1NN evolved as a common baseline [63]. Multiple evaluations showed that 1NN is hard to beat in time series classification, especially when combined with the DTW measure [23, 274]. For large training data sets, it was shown that the predictive quality with euclidean distance approaches that of elastic measures such as DTW [264]. Unfortunately, kNN suffers from high memory costs and long prediction times as all training examples have

to be stored (both $\mathcal{O}(NT)$ for training set size N and signal length T in a naive implementation). To make a prediction on a new time series, the DTW measure has to be computed for all these training examples, resulting in high computational demands and long prediction times. In [187], nearest centroid (NC) combined with DBA was shown to be competitive with kNN at a much smaller computational cost (i.e., prediction time) and reduced memory space demand across multiple data sets. This was confirmed in [63] for barycenter averaging with the Soft-DTW measure. NC methods rely on each anomaly class being sufficiently representable by a single centroid. For a binary anomaly detection, finding outliers can then be approached by comparison of test signals to the normal class centroid.

Multi-dimensional representation based methods Feature space methods yield a powerful way to reduce the information given by raw samples in sensor signals. As mentioned above, these approaches come with the challenge to identify a sensible set of features when aiming for a generic anomaly detection: For a generic anomaly detection, it is typically infeasible to specify the most relevant features a priori. Thus, a potentially large set of features has to be computed. As discussed in [7], high-dimensional feature spaces result in increasing distances between all data points, which makes common approaches of finding anomalies by large distances to normal data points or in regions with a small density of data points increasingly less appropriate. This is known as the curse of dimensionality and was described first in [9] for applications of high dimensional outlier detection. Thus, feature space approaches in anomaly detection have to come with an implicit or explicit feature selection (e.g., decision tree based approaches), dimensionality reduction (e.g., subspace methods) or have to be robust to irrelevant features and the high dimensionality of the feature space (e.g., robust covariance estimators [205]). The challenge of defining the most relevant features a priori for feature space based methods might alternatively be circumvented by relying on feature learning techniques. Apart from sparse dictionary techniques like non-negative matrix factorization (NMF), neural network based methods dominated the field of feature learning. Despite their dominance in image classification, their application in time series anomaly detection fields is rather seldom.

Purely unsupervised, multi-dimensional anomaly detection methods model anomalies as outlying points from dense regions of data points [129]. Dense normal regions can be identified either by model-based approaches like one-class classifiers [207, 220, 249] and probabilistic models [205] or proximity-based approaches. The latter group of algorithms can further be distinguished into distance-based methods (often kNN-based approaches like ODIN [94]) and density-based approaches like LOF [46] and its extensions [96, 224, 248]. Other popular proximity-based approaches are INFLO [113], LoOP [128], LDOF [286], LDF [135] and KDEOS [223]. More advanced, hierarchical density-based approaches were introduced by DBSCAN [72] and extensions like OPTICS [17] or HDBSCAN [49].

Many of the former methods relied on data being given as complete batch, i.e., data have been considered in an offline classification scenario. Recently,

the streaming data scenario (i.e., online classification) received more attention triggered by the position papers of Aggarwal [6] and Zimek [297]. Dominant techniques relied on ensemble methods based on the early success of isolation forests [146] and the theoretical analysis of anomaly ensembles in [8]. Recent work on outlier ensembles in streaming data scenarios is listed in [157] and given by the subsampling techniques in [296], ensembles of randomized space trees [272] or half-space trees [247], selective [198] and sequential [199] anomaly ensembles, histogram-based ensembles like LODA [188] and subspace hashing ensembles like RS-Hash [215] or xStream [157].

2.4.2 Deep Models

Deep anomaly detection models can be distinguished regarding the representation used as input for the model, regarding the architecture of the model and the anomaly detection loss. Here, any neural network with more than one hidden layer is considered a deep model.

Input Representations for Deep Anomaly Detection Deep learning based methods are most prominently applied in image modeling tasks. In general, two-dimensional representations open up perspectives for making use of the numerous methods applied in this field. For time series data, this involves finding a reasonable two-dimensional embedding. Despite embeddings based on Gramian angular fields [265] or Markov transition fields [48, 265], the field of 2D time series embeddings is dominated by TFDs: Starting in applications for acoustic modeling in speech recognition [99] and speech generation [105, 106], TFDs started to enter the field of MHM applications [147]. Compared to more traditional approaches in MHM often building on computationally efficient (e.g., statistical) features (cf. Section 2.1), TFDs introduce a non-negligibly larger additional running cost during prediction time.

Architectures for Deep Anomaly Detection Among deep models based on one-dimensional time series embeddings, approaches based on RNNs [155, 285] dominated the field over the course of many years [24]. Due to the problem of vanishing gradients [28], long short-term memory (LSTM) [102] and gated recurrent unit (GRU) architectures [58, 132] replaced simpler RNN architectures to a large extent. More recently, non-RNN approaches, e.g., based on VAEs [275] with MLP layers started to gain interest. Especially the advent of temporal convolutional networks (TCNs) [24] led to the belief that RNNs should not be considered without any alternative in the field of time series modeling tasks.

In [24], Bai et al. introduced a generic temporal convolutional network (TCN) architecture which they applied to a wide range of sequence modeling tasks. They found, that TCNs often outperformed diverse RNN architectures (state of the art LSTMs and GRUs among others) and suggested replacing RNN architectures as the canonical starting point for sequence modeling tasks (like time series modeling) by TCNs. They justified this suggestion not only by the better predictive performance but multiple other advantages like the TCN-inherent

parallelism for processing of sequences both during training and prediction, the flexibility in controlling the models' effective memory (i.e., the TCN's receptive field), the better stability of gradients during learning (compared to the vanishing gradients problem for many RNNs architectures [28]) and the smaller memory requirements during training compared to RNNs [24]. On the downside, they argued that TCNs might possibly expose higher memory requirements during evaluation and come with a higher dependency of the necessary memory size (i.e., receptive field) on the application domain. Extending TCNs to be competitive to stochastic RNN architectures [27, 88] is part of ongoing research and dealt with in [13, 134] among others.

Loss Functions for Deep Anomaly Detection Regarding loss functions for deep anomaly detection models, autoencoder (AE) approaches dominated the field for many years [207], whereas other unsupervised representation learning methods like GANs [86] are rarely applied [218]. AEs are constructed from an encoder and decoder part and try to learn the identity function (i.e., try to learn an input-output mapping which allows for the reconstruction of the input with minimal reconstruction error) [207]. The encoder-decoder structure paired with a reduced intermediate dimension or sparsity regularization yields a computational bottleneck in the middle part of the network which forces the network to extract a meaningful set of latent features. When the training of the model succeeds, it is capable of learning the latent features necessary for explanation of normal data in order to reconstruct them properly. Anomalous data samples however, which are outliers with respect to this normal data, are expected to expose data variations not fully captured by this latent intermediate representation, thus yielding a higher reconstruction error $\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2$ between input $\mathbf{x}$ and reconstruction $\hat{\mathbf{x}}$. The learned intermediate embedding can either be used as plug-in feature embedding for classical anomaly detection approaches (cf. Subsection 2.4.1) or in a fully deep anomaly detection by applying the reconstruction error as anomaly score [207].

Extensions of classical AE approaches in anomaly detection include denoising AEs [257, 259], sparse AEs [153], AE approaches leveraging additional information from the latent representation for construction of an anomaly score [298] and VAEs [124]. Although these AE-based anomaly detection approaches are predominant in image-based modeling and thus favor two-dimensional data representations, especially VAEs were more recently applied to time series data in direct or other one-dimensional embeddings [12, 114, 275]. Typically, VAE-based anomaly detection models rely rather on the reconstruction probability than on the reconstruction error, which is more independent from the characteristics of a given data set [12].

Both GANs-based and (V)AE-based approaches rely on reconstruction measures as anomaly scores. Although the assumption that anomalies might reflect in a higher reconstruction error or reconstruction probability, neither of them represent a loss function custom-built for anomaly detection. On the other hand, anomaly detection based on one-class neural network loss functions are a promising direction for deep anomaly detection which come with a tailor-made anomaly

detection loss function [52, 207]. In this thesis, the Deep SVDD approach introduced in [207] is focused on, due to being theoretically well-founded and empirically approved by its shallow counterparts (OC-SVMs, SVDDs) over the course of many years. The approach is closely related to existing OC-SVM [220] and SVDD [249] approaches but comes without the necessity of handcrafted feature engineering and other drawbacks related to construction and manipulation of the kernel matrix (like at least quadratical computational scaling in the number of samples) and its nonparametric nature (potentially large number of support vectors to be kept permanently in memory) [207]. In addition, the authors most recently proposed a semi-supervised extension to the Deep SVDD approach (which they term Deep SAD) showing promising results even when being provided with only a small number of labeled training data examples [208].

2.5 Annotations by Human Users

A vast amount of literature on the specifics of human annotations exists. Among the most typical application fields are medical applications like smoking detection [3], sleep detection [4] or affect recognition [233] and the large field of activity recognition [163, 261]. While earlier work focused on collecting labels from diaries filled out by study participants, smartphone apps took over the field of human annotation [59, 62, 254, 284]. The main advantage of collecting labels online via smart phones is timely labeling triggered by events (e.g., from sensor data) paired with visualization and accessibility of context data in order to give the user a sensible amount of information during annotation.

Online annotation of sensor signals in industrial manufacturing surroundings was so far not yet considered in the research community. Thus, for discussion of state-of-the-art online annotation methods in this section, other, related application fields are considered.

In [189], authors proposed a procedure for the synchronization of wearable accelerometers and video cameras for automatic ground truth annotation of acceleration sensor signals. This allowed them to estimate time delays between these two sensor modalities with a minimal level of user interaction and thus improve the annotation of acceleration sensor signals via video footage inspection.

Authors of [165] proposed an online active learning framework to collect user-provided annotations, as opposed to the typical retrospective analysis of video footage used in human activity recognition (HAR). Only highly critical annotations were prompted to the user, which is similar to the live annotation approach in Chapter 5 which prompts only anomalous signals for online annotator feedback.

In examining their results, Miu et al. claimed that users of activity recognition systems themselves are sources of ground truth labels that are often neglected [165]. This makes sense for the field of activity recognition, where users have a good knowledge of their own activities. For MHM applications as considered in this thesis, it is less clear in advance if human annotators (i.e., machine operators) have a good knowledge of the current machine behavior such

that they are reliable annotation sources and their labels can be considered as ground truth.

In [221], Schroeder et al. performed an analysis of existing live annotation systems and suggested then an own online annotation system based on their findings about basic requirements for annotation systems. This online annotation system was generated automatically based on a database schema. In addition, their setup allowed to include annotation constraints, which can be used for causal correction of given annotations.

In [164], Miu et al. assumed the existence of a fixed, limited budget of annotations a user is willing to provide and discussed different strategies for best spending this budget. This is related to the assumptions described in later sections, that the quality of human annotations relies both on the quality of signals proposed as abnormal to the user (i.e., small false positive (FP) rate) and (visual) clarity of anomalies prompted to the user for annotation.

In [84], authors proposed a technique for online activity discovery based on clustering assumptions of labels in successive signal windows. Although their approach is memory efficient and has constant time complexity, it is not applicable for live annotation of signals as proposed in later sections due to the fact, that reoccurring activities lead each time to a newly created cluster segment. This does not allow to model normal behavior as a single class in reoccurring cluster segments and distinguishing it from other, abnormal signal classes. This is however crucial for the live annotation approach relying on prompting only outliers from this single normal signal class for user annotation.

In general, the authors of [165, 189, 221] showed that online annotation by user feedback can yield comparable or even better results to retrospective annotation (e.g., via video footage), even when considering a fixed budget of annotations [164]. This is reasonable for the typically considered task of human activity recognition, where the user is an expert for his own activities. For the task of detecting different types of machine health anomalies, it is a priori less clear if and for which anomaly classes the human annotators (e.g., machine operators) can be considered experts yielding a reliable ground truth labeling.

2.6 Weakly Supervised Learning

Most performant models in machine learning across a wide range of predictive tasks are dominated by supervised approaches. The performance of these methods increases typically with the amount and quality of given labels. However, large sets of high-quality labels come with non-neglectable costs due to an additional time effort spent on annotation. In order to obtain a high quality, the process of creating label sets typically involves expensive domain expert annotations. Thus, supervised techniques come with an inherent quality-cost trade-off due to the annotation process: Additional annotations are expensive.

In order to reduce the cost of annotations, different strategies were proposed. These were recently summarized by the umbrella term *weakly supervision* and are thus explicitly contrasted to the traditional strong supervision techniques (i.e.,

providing complete sets of high-quality ground truth labels for the set of data points). Weak supervision techniques can be classified as follows [292]:

- *Incomplete supervision*: The core idea of this class of weak supervision approaches is reducing the amount of annotated data instances. The two most prominent representatives of this class are semi-supervised learning [54, 295] and active learning [226]. Semi-supervised learning aims at combining large amounts of unlabeled data with a smaller subset of labeled data instances during the model learning process. Active learning aims at finding an optimal strategy for selecting only the most valuable data instances to be proposed for annotation. High value can be defined, among other strategies, by high uncertainty of the predictive model regarding classification of the given data (which is referred to as uncertainty sampling [139]) or by the scarcity of assumed labels (i.e., rare labels are more valuable).

- Similar to the former class of methods, *inexact supervision* methods reduce the amount of annotations. Other than incomplete supervision methods, inexact supervision groups data instances and then queries for annotations of these groups, resulting in a more coarse-grained annotation [292]. The most typical approaches are multiple instance learning [76, 158] and label proportions learning [98]. Multiple instance learning groups data instances into bags and queries labels for these bags, thus effectively reducing the labeling effort. Typically, a bag of data instances is labeled as positive when at least one instance in the bag is considered positive [292]. Label proportions learning additionally assumes knowledge about the proportions of label classes in a bag and uses this information for supervision.

- Finally, *inaccurate supervision* provides complementary methods to the former two groups of incomplete and inexact supervision: In contrast to reducing the amount of data instances that are proposed for annotation, inaccurate supervision decreases labeling costs by tolerating a higher amount of erroneous labels, thus effectively reducing the quality of annotations by introducing more noise into the labeling process. A typical reason for the higher amount of label noise is the waiver of domain expert labeling sacrificed in order to reduce the cost of annotation. A typical alternative to domain expert labeling is given by crowd source annotations [44]. As crowd sourcer annotations are known to be of lower annotation quality and thus introduce a higher amount of noise into the labeling process, many of the proposed approaches try to estimate reliability of the label feedback. When no ground truth labels are present, the most typical strategy is rating reliability by inter annotator agreement. Despite several proposed statistical measures [90], approaches leveraged label proportions [194], Bayesian nonparametric estimators [166] or adversial models [279] for estimating label reliability from inter annotator agreement. In addition to techniques for label reliability estimation and denoising, inaccurate supervision techniques necessitate an appropriate design of annotation protocols and interfaces [292].

Label reliability estimation and denoising techniques aim at revealing the latent underlying true labels from the observed noisy labels, often making use of additional knowledge about domain-specific characteristics of the label structure. For example, time series label sequences often exhibit a low-varying and/or piecewise stationary structure. Temporally noisy positive labels are thus considered to appear close (in a temporal sense) to the true positive examples [3]. Then, label denoising can be approached by methods like alignment of the label vector with class scores predicted by the model, either in a single instance (i.e., align single labels with data), multiple instance (i.e., align bags of labels with data) or label proportion setting (i.e., use information of fraction of positive labels in a bag during alignment with data). Other approaches leverage assumptions about the non-iid structure of temporal data in order to extend classical iid classifier or outlier detection models [87]. A more elaborate denoising approach for temporally adjacent instances is stated in [3], where the information about order and positions of positive instances is retained and used during label denoising (as opposed to multiple instance and label proportions approaches stated above).

Another group of methods learns user-specific models, in order to more precisely capture the subjective labeling behavior of annotators and possibly sort out "spammers" (i.e., annotators with a close-to-random labeling behavior) or "adversaries" (i.e., annotators deliberately assigning incorrect labels) [292]. Several approaches like proposed in [103, 182, 237] explicitly estimated user-specific reliability models or tried to improve the annotation quality by imposing additional assumptions on the characteristics of labels (e.g., correlations between adjacent labels [3, 4]).

Finally, non-expert labels can be synthesized from weak information sources by means of a generative model. Label generative models can be represented efficiently by a probabilistic graphical model (PGM) [21]. These allow to learn the dependencies between weak information sources like domain heuristics, features, weak classifiers or non-reliable human annotators, and estimate more reliable labels by combining the weak information obtained from these dependent sources. In data programming settings [196], such weak information sources are typically termed *labeling functions (LFs)*. Learning the reliability of the label-generating LFs is a problem which has been tackled by a variety of approaches in supervised settings [161, 197]. In unsupervised settings however (i.e., when true labels are never observed), the problem of structure learning of graphical models becomes more challenging, as the true labels have to be modeled as latent variables [21]. In order to reduce the complexity of the structure learning task and circumvent approximations for the gradient of the learning objective (e.g., Gibbs sampling or variational methods), simplifications like assuming conditional independence of the labeling functions or maximizing a pseudo-likelihood instead of the marginal likelihood [21] are popular. Chapter 6 presents a more detailed discussion on these topics.

2.7 Summary

- In Section 2.1, related work in the diverse field of MHM applications was summarized. A focus was put on tool condition monitoring (TCM) and imbalance detection. Throughout this thesis, related work is extended by several contributions. First, a novel TCM health indicator is proposed. Furthermore, methods for automated detection of imbalances in different rotating machine parts (grinding wheel, dressing tool, etc.) are described. Finally, the fact that in condition-based monitoring systems context information regarding process adaptations, machine part changes and anomalous events is often neglected was discussed. This concern is addressed by the prototypical labeling system developed for the live annotation proof-of-concept study to be presented in Chapter 5, which allows annotating such events by domain experts parallel to recording of measurement data.

- Section 2.2 summarized different methods for segmentation of sensor signals. Changepoint approaches were emphasized in this summary due to assumptions imposed by these methods fulfilled by the nature of the sensor data in this thesis. It was pointed out however, that no computationally efficient method incorporating the cyclostationary structure of this sensor data exists. In this thesis, an empirical estimator of generic cyclostationary data structures is presented which addresses this research gap. The recurrent segment borders found by this approach can be used both to detect several suddenly occurring anomalies in the production process of machine tools and for definition of comparable signal regions in which health indicators for anomalies with a drifting character (e.g., the TCM health indicator mentioned in the former list item) are extracted.

- In Section 2.3, methods for recovery of discrete frequency components from time frequency distributions computed for the sensor data were discussed. Knowledge of this discrete frequency components is used in this thesis for condition monitoring of specific rotating machine parts (e.g., imbalance detection of rotating machinery mentioned in the first item of this summary list).

- Section 2.4 gave an overview over several generic anomaly detection approaches, both deep architectures and shallow models. Different shallow models are compared in this thesis in order to select an appropriate model for the live annotation approach presented in Chapter 5. Furthermore, several neural architectures are compared for detection of anomalies in data recorded during conducting the proof-of-concept study of this live annotation approach.

- In Section 2.5, specifics of human annotations were discussed, with a focus on disadvantages and advantages of online annotation in activity recognition applications. The basic idea of annotating hard-to-interpret raw (acceleration) sensor signals by considering human-interpretable meta information (e.g., video footage) is adopted for the live annotation approach

in that direct, human-interpretable context/meta information is presented (e.g., being able to view and hear the processing of workpieces) while being given sensor signals for review.

- Section 2.6 gave an overview of different weak supervision methods. The Deep SAD loss function [208], which is applied in this thesis for anomaly detection with neural architectures, is extended to include probabilistic labels in Chapter 6. These probabilistic labels are obtained by weakly supervised approaches.

Part I

Task-Specific Machine Monitoring Features and Models

The features stated in the related work section 2.1 are mostly general purpose features which proved useful in a wide range of industrial applications but are not specific for a certain prediction task at hand. In addition, phenomena like sensor drifts, user-initiated adjustments of process parameters and changes of workpiece types that are not related to the actual prediction task influence or even dominate the scores of such generic features. Consequently, when the sensor data used for training of models illustrates different process parameter settings or workpiece types than in-field recorded test data on which these models are executed on, a reduction of the predictive expressiveness of these features and thus predictive quality occurs.

In the following part of this thesis, it is demonstrated how to include domain expertise into the process of engineering features for selected predictive tasks that are less affected by the covariate shift problem described above: The benefit both of signal segmentation (Chapter 3) for detection of sudden process-related anomalies and tool condition monitoring, as well as estimation of discrete frequencies for condition monitoring tailor-made to certain machine parts (Chapter 4) is outlined. Extensions on existing approaches for signal segmentation are introduced, the applicability of estimation and tracking of discrete frequencies for condition monitoring purposes is discussed and tailor-made features for specific MHM tasks are suggested.

3

Signal Segmentation

In this chapter, various methods for segmentation of time series in machine condition monitoring and process monitoring are discussed and compared regarding their performance on the measured sensor data. In Fig. 3.1, exemplary data recorded from a vibration sensor mounted at the workpiece support during machining a workpiece are illustrated.

The depicted sensor data stream illustrates a hierarchical recurrent structure, i.e., exhibits repetitive patterns on various levels: Firstly, the data stream can be segmented into four recurrent high-level data records (depicted by different color shadings in Fig. 3.1). Each of these four data records is related to the machining of a single workpiece. Identifying this high-level recurrent structure is not in focus of this thesis, as trigger signals are provided that allow deterministically subdividing the data stream into records related to the machining of single workpieces. Thus, the data sets in this thesis consist of data records, each of which is related to the machining of a single workpiece. These data records are referred to as signals in the following. Furthermore, a second, low-level recurrent structure can be identified in each of the four data records related to the machining of a single workpiece. This low-level recurrent structure originates from the same sequence of processing steps applied during machining of each workpiece with a profiled grinding wheel: First, the grinding wheel approaches the workpiece. After initial contact of grinding wheel and workpiece, successive processing steps are performed (roughing, finishing, etc.). While the amount of material removal is defined by the processing step, locations of material removal are defined by the profile of the grinding wheel. The low-level recurrent structure caused by the repetitive sequence of processing steps and machining with a profiled grinding wheel can consequently be identified in sensor nodes attached to process-related machine parts. The sensor node attached to the workpiece support depicts these process-related specifics of the signals.

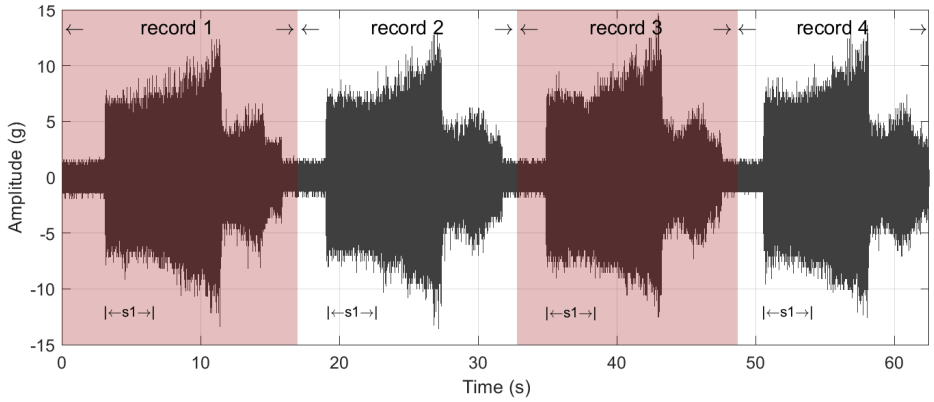


Figure 3.1: *Hierarchical recurrent structure of sensor data, recorded with a sensor attached to the workpiece support of a grinding machine. The colored shading highlights the high-level recurrent structure of data records (machining of four workpieces). Each high-level record illustrates a low-level recurrent segment structure (similar segments due to same processing steps applied during machining of each workpiece), illustrated exemplary for the recurrent segment* $s1$.

Various methods and benefits for automatically identifying this low-level recurrent segment structure in sensor signals are discussed in the following chapter. In Section 2.2, an overview over the wide field of signal segmentation methods was presented, including piecewise linear approximation (PLA) methods, clustering models (GMM and HMM) and changepoint detection methods. PLA based methods and likelihood ratio based changepoint methods are excluded from the detailed comparison. This is due to empirical evaluations in a master's thesis [172] supervised by the author, suggesting both groups of methods not to be suitable for the sensor data. Clustering-based models and Bayesian changepoint detection approaches (with a focus on the Bayesian Online Changepoint Detection (BOCPD) algorithm [5]), being best suited for the nature of the measured sensor data, are discussed deeper in Section 3.2.

A computationally convenient approximation of HMM-based signal segmentation is presented by combining GMMs and FSMs (Subsection 3.2.1). In Subsection 3.2.2, an extension to the BOCPD algorithm is introduced. This extension which was termed changepoint recurrence distribution (CPRD) allows estimating recurrent signal segments more robustly than via standard BOCPD and can additionally be used for a computationally efficient detection of process-related anomalies.

In Section 3.3, signal segmentation by HMMs, GMMs and BOCPD is compared both regarding quality and computational cost of signal segmentation. Furthermore, the capability of the presented CPRD extension to distinguish changepoints provoked by the repetitive processing steps from changepoints caused by process-related anomalies is discussed. In addition, a novel tool condition health

indicator relying on recurrent segments identified by the proposed CPRD extension is introduced. Finally, conclusions regarding the optimum choice of the signal segmentation method are drawn in Section 3.4.

3.1 Motivation

Automatically identifying recurrent segments in the sensor data is of high interest for two reasons:

1. In recurrent low-level segments, similar processing steps result in stationary statistical behavior of the data-generating process (cf. Fig. 2.4). Thus, recurrent segments in successive records exhibit similar statistics, i.e., cyclostationary behavior. Modeling this cyclostationary behavior allows extracting comparable feature scores for detection of **drifting anomalies**. This is illustrated exemplary for the prediction task of tool condition monitoring (TCM).

2. In addition, being able to describe normal behavior of the processing step sequence allows for detection of deviations caused by **suddenly occurring anomalies** that manifest in changes in the normal segment structure. This is illustrated for the exemplary tasks of detecting abnormal machine part contacts and signal form deteriorations (here due to a grinding wheel imbalance).

Identifying recurrent segments in a data stream is a non-trivial task as outlined in the following chapter. As discussed in Section 2.2, different approaches based on piecewise linear approximation of signals, clustering-based models and algorithms involving a penalized likelihood function of the data are among most popular choices. With the BOCPD approach introduced in [5], a new direction based on recursive Bayesian estimation of signal changepoints emerged. These changepoints happen to occur at points of significant changes in the statistical parameters of the underlying data-generating process. During normal processing behavior, changepoints tend to coincide with processing step changes. However, changepoints additionally occur due to abnormal machining behavior (e.g., abnormal machine part contacts) or signal fluctuations.

In order to identify recurrent signal segments more robustly and being able to distinguish between normal (recurrent) and abnormal (non-recurrent) changepoints, an estimator for the probability of changepoints to be recurrent is introduced. The dependence of the probability of changepoint recurrence on the relative position of the changepoint in the signal allows defining a distribution of probability of changepoint recurrence across the signal length. The estimate of this *changepoint recurrence distribution (CPRD)* can then be used to incorporate the domain knowledge of the data's recurrent structure into a more informative (i.e., non-uniform) prior distribution of the BOCPD changepoint estimator or in a separate step on classifying BOCPD changepoints regarding their probability to be recurrent.

3.2 Methods

In this section, methods which are applied for signal segmentation in the upcoming experiments are outlined in more detail than in Section 2.2. These methods can be categorized into two groups, clustering-based signal segmentation and Bayesian changepoint detection.

Clustering-based approaches (GMMs, HMM) assign a most probable cluster membership estimate to successive data samples. Segmentation of the data sample stream can then be obtained by finding transitions in these successive cluster membership estimates. Typically, the original time series data is transformed to a dual space by extraction of features from these time series. This is motivated by a better observability of clusters in this feature space than in the original time series data. Then, a clustering model of appropriate complexity has to be trained, where complexity refers to the number of clusters assumed by the model. Thus, the quality of signal segmentation relies both on a choice of appropriate features and model selection (i.e., estimation of the optimal number of clusters). For GMM model selection, information-theoretical criteria AIC and BIC are compared. In addition, an FSM for post-processing of GMM cluster membership estimates is proposed. FSM post-processing allows to constrain cluster estimates to follow a certain temporal order and thus mimic the behavior of an HMM but at reduced computational effort.

For Bayesian changepoint detection, the generic BOCPD changepoint estimator is extended in order to capture the specific, recurrent nature of the sensor data in this thesis. For this, the CPRD estimator of changepoint recurrence is proposed. Additional to a robust estimation of signal segments, the CPRD allows for a computationally efficient unsupervised anomaly detection.

3.2.1 Modeling Recurrent Signal Segments with Gaussian Mixture Models

HMMs are widely applied in time series applications and generally appealing for identifying the structure of time series. Their capability to model latent, hidden states that generate the observed data samples allows to infer the data-generating process and its cyclostationary behavior in a powerful way: Recurrent segments can conveniently be defined as piecewise constant state sequences in the Viterbi path (i.e., the estimated most likely sequence of hidden states). However, HMMs can be computationally challenging on resource-constrained embedded systems: Parameter learning of HMMs is computationally expensive but has to be performed only once. Time-critical online prediction (i.e., computing the Viterbi path) with the learned HMM however involves solving a dynamic programming problem per each sensor signal.

GMMs are computationally less expensive and model the state memberships by probabilistic means similar to HMMs. However, they do not encompass the temporal dependency of successive signal samples. Thus, after predicting most likely cluster membership with the GMM, postprocessing these cluster estimates by an additional L2R FSM is proposed to mimic an HMM's structure.

The proposed algorithm for signal segmentation via a combination of GMMs and FSM is summarized in Algorithm 1. Based on extraction of M features, $M \times 1$-dimensional observation vectors $\mathbf{o}_{i,1}, \ldots, \mathbf{o}_{i,T}$ per each of T successive fixed length blocks of data records rec_i are computed. These T observation vectors are concatenated in $M \times T$-dimensional observation matrices $\mathbf{O}_i$ per data record rec_i. This is repeated for N training data records rec_i.

Afterwards, K GMM model candidates $\boldsymbol{\theta}_j = \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K$ are learned based on the training observation matrices $\mathbf{O}_1, \ldots, \mathbf{O}_N$. For GMMs, these models are parameterized by $\boldsymbol{\theta}_j = \{\boldsymbol{\pi}_{1:j}, \boldsymbol{\mu}_{1:j}, \Sigma_{1:j}\}$, where $\boldsymbol{\pi}_{1:j}$ are the $j \times 1$ mixture proportions, $\boldsymbol{\mu}_{1:j}$ consists of j $M \times 1$ mean vectors and $\Sigma_{1:j}$ are j $M \times M$ covariance matrices for each of a mixture of j normally distributed clusters [168].

Among these model candidates $\boldsymbol{\theta}_j$, model selection is performed via a predefined information criterion IC. In this thesis, the Akaike Information Criterion (AIC) [11] and Bayesian Information Criterion (BIC) [225] as the most common criteria for GMM model selection [60] are compared. Both criteria consist of two terms: a goodness of fit term and a term penalizing model complexity. While AIC measures complexity only by the number of model parameters q, BIC incorporates the sample size into the penalty term:

$$AIC_j = -2 \log L(\boldsymbol{\theta}_j; \mathbf{O}) + 2q \tag{3.1}$$

$$BIC_j = -2 \log L(\boldsymbol{\theta}_j; \mathbf{O}) + q \cdot \log(N \cdot T) \tag{3.2}$$

where $L(\cdot)$ represents the likelihood function and $\boldsymbol{\theta}_j = \{\boldsymbol{\pi}_{1:j}, \boldsymbol{\mu}_{1:j}, \Sigma_{1:j}\}$ is the set of learned parameters for a GMM with j clusters. The sample size is given by $N \cdot T$, i.e., the product of the number of training data records N times the number of fixed length blocks T per data record rec_i.

The optimal model is typically found either at the minimum IC value or the "knee point" [287]. For a more detailed explanation of knee point computation the reader is referred to [287]. Put in a nutshell, knee point detection identifies the optimal model candidate at the point where the decrease of successive IC values starts to flatten, i.e., the point where the dominance of the negative log likelihood term $-2 \log L(\boldsymbol{\theta}_j; \mathbf{O})$ vanishes in favor of the regularization term $q \cdot \log(N \cdot T)$. Assigning the optimal model at this knee point can perform beneficial to simple assignment at minimal IC values: Depending on the sample size $N \cdot T$, IC values might decrease across all model candidates $\boldsymbol{\theta}_j$, thus favoring the most complex model. Finally, the optimal matching hidden state vector $\mathbf{z}_i$ for observation matrices $\mathbf{O}_i$ is identified with this optimal model $\boldsymbol{\theta}_{sel}$ by computation of posterior probabilities of the mixture components.

Both information criteria AIC and BIC identify the GMM explaining feature score observations $\mathbf{o}_{i,t}$ in the most sensible way according to the assumptions made by their formulation as regularized maximum likelihood estimators. However, neither of both criteria allows identifying the temporal order of these clusters. Furthermore, the following step of GMM cluster membership estimation does not incorporate temporal information inherent to successive observations $\mathbf{o}_{i,t}$. When aiming for segmentation of original time domain data records rec_i by finding transitions in successive cluster membership estimates $\mathbf{z}_{i,t}$, not incor-

porating the temporal meta information during cluster membership estimation might favor estimating insensibly many cluster transitions. This results directly in an over-segmentation of the data records rec_i. HMMs, on the other hand, use this temporal meta information of successive observations $\mathbf{o}_{i,t}$ in order to constrain cluster transitions to follow a feasible temporal order of clusters. This reduces the risk for over-segmentation of signals due to erroneous assignments of cluster transitions.

The following approach is proposed in order to address these shortcomings of a purely GMM-based signal segmentation. First, the temporal order seq_j of GMM clusters of $\boldsymbol{\theta}_{sel}$ has to be identified. This is done by computing the median $\tilde{t}_{i,j}$ of block indices t associated with GMM clusters j per each hidden state vector $\mathbf{z}_i$ and then again computing the median indices $\tilde{t}_j$ across all N training records. This allows to identify the temporal order seq_j of the GMM clusters by sorting them regarding their median block indices $\tilde{t}_j$. Then, an L2R FSM can be defined using these temporally ordered GMM clusters as states. This L2R FSM mimics the effect of an L2R HMM as illustrated in Fig. 2.2 in that it assigns GMM clusters to the temporally ordered FSM states $s_1, \ldots, s_5$ (representing the deterministic sequence of processing steps). The L2R FSM can then finally be used to post-process the GMM hidden state vectors $\mathbf{z}_i$ by constraining transitions between GMM clusters to follow a temporal L2R order. Thus, the L2R FSM reduces the space of admissible state transitions for the optimal GMM $\boldsymbol{\theta}_{sel}$, finally allowing to find a temporally constrained estimate $\mathbf{z}_{FSM,i}$ of the GMM hidden state vector $\mathbf{z}_i$.

3.2.2 Bayesian Estimation of Recurrent Signal Segments

Disclaimer: Parts of this section were taken verbatim from own previous publications ([202, 203] ©2019 IEEE).

Bayesian changepoint estimation via BOCPD relies on fundamentally different principles to identify signal segments than the clustering approaches illustrated in the previous section. One of the main advantages is that BOCPD does not necessitate an explicit and static model selection. Instead, BOCPD updates sufficient statistics of the data-generating process model directly based on assumptions specified about the data structure in the prior distribution $p(\mathbf{x})$ and assumptions concerning the occurrence of changepoints specified in the conditional run length prior distribution $p(r_t|r_{t-1})$. In the original BOCPD approach as introduced in [5], the quite conservative assumption of probabilities of changepoint occurrence being independent from former changepoints and distances between them is made. This assumption on the structure of the conditional prior $p(r_t|r_{t-1})$ is imposed by choosing the uninformative constant hazard function $H(\tau) = 1/\lambda$ which was discussed in Section 2.2. In the following section, a straightforward and effective way to empirically estimate a changepoint prior from the run length distribution $p(r_t|\mathbf{x}_{1:t})$ is introduced. This so-called CPRD extension is more tailor-made to the characteristic, recurrent structure of MHM data.

Throughout all experiments conducted for signal segmentation via Bayesian

Algorithm 1: GMM+FSM signal segmentation model

Input: N training sensor data records $rec_1, \ldots, rec_N$
Feature functions $\Phi_1, \ldots, \Phi_M$
Type of information criterion IC
Upper bound K on number of states
Output: Hidden state vectors $\mathbf{z}_{FSM,i}$

for each rec_i **do** ▷ Construct observation matrices $\mathbf{O}_1, \ldots, \mathbf{O}_N$
  for each $block_t$ in rec_i **do**
    $\mathbf{o}_{i,t} = [\Phi_1(block_t), \ldots, \Phi_M(block_t)]^T$
  end for
end for
for a max. number $j = 1, \ldots, K$ of GMM states **do**
  $\boldsymbol{\theta}_j \leftarrow$ GmmTrain$(\mathbf{O}_{1:N}, j)$ ▷ Train GMM with j components
  $IC_j =$ InformationCrit$(\boldsymbol{\theta}_j, N)$
end for
$\boldsymbol{\theta}_{sel} \leftarrow$ KneePoint$(\boldsymbol{\theta}_{1\ldots K}, IC_{1\ldots K})$ ▷ Select best-fit GMM model $\boldsymbol{\theta}_{sel}$
for each rec_i **do** ▷ Temporal ordering of GMM clusters
  $\mathbf{z}_i \leftarrow$ GmmPredict$(\boldsymbol{\theta}_{sel}, \mathbf{O}_i)$ ▷ Infer hidden state vectors $\mathbf{z}_i$
  for all clusters j of GMM $\boldsymbol{\theta}_{sel}$ **do**
    $\tilde{t}_{i,j} \leftarrow$ Median(SelectIndices$(z_{i,t} == j)$)
  end for
end for
$\tilde{t}_j \leftarrow$ Median$(\tilde{t}_{i,j})$
$seq_j \leftarrow$ Sort$(\tilde{t}_j)$
$\boldsymbol{\theta}_{FSM} \leftarrow$ FsmTrain(j, seq_j) ▷ Train L2R FSM
$\mathbf{z}_{FSM,i} \leftarrow$ FsmPredict$(\boldsymbol{\theta}_{FSM}, \mathbf{z}_i)$ ▷ Constrain temporal order of $\mathbf{z}_i$
return $\mathbf{z}_{FSM,i}$

changepoint estimators, a simple signal representation of the raw sensor samples is computed from average rectified valuess (ARVs). Average rectified values are computed by the mean of absolute values in non-overlapping fixed blocks of M raw samples y_t, i.e., $\frac{1}{M}\sum_{i=1}^{M}|y_i|$ in each successive signal block comprising $M = 1024$ raw data samples y_i. These ARV representations are referred to as envelope signals in the following. The main reason for this is the piecewise unimodal distribution of envelope signal samples which can effectively approximated by normal distribution. This was illustrated in Subsection 2.2.3. Furthermore, the decimation induced by the envelope extraction results in a reduced computational effort for signal segmentation.

Changepoint Recurrence Distribution (CPRD)

This section presents the CPRD estimator on changepoint recurrence. In the BOCPD algorithm this CPRD estimator builds upon, changepoints are identified from the run length distribution $p(r_t|\mathbf{x}_{1:t})$. Due to the typical concentration of

probability mass of the run length distribution at a dominant peak, the most probable run length estimate $\hat{r}_t$ can be approximated sensibly at the maximum a posteriori (MAP) estimate of the run length distribution, i.e.,

$$\hat{r}_t = \arg\max_{r_t} p\Big(r_t|\mathbf{x}_{1:t}\Big) \tag{3.3}$$

According to [5], changepoints can be assigned at $\hat{r}_t = 0$. However, for machine tool data with potentially smooth transitions between signal segments, changepoints at these segment borders do not necessarily lead to $\hat{r}_t = 0$, but to a major drop in this most probable run length estimate $\hat{r}_t$. Drops in $\hat{r}_t$ (i.e., where $\hat{r}_t$ does not increase by one) can then be interpreted as changepoints with a non-zero changepoint probability

$$p\Big(c_t|\mathbf{x}_{1:t}\Big) \triangleq p\Big(\hat{r}_t|\mathbf{x}_{1:t}\Big)\Big|_{\frac{\partial \hat{r}_t}{\partial t}\neq 1} \tag{3.4}$$

where $\frac{\partial}{\partial t}$ denotes a derivative with respect to t. Changepoints c_t occur not necessarily due to recurrent changes of process steps, but can also be due to signal fluctuations or anomalies. This motivates the necessity to separate recurrent changepoints from the set of all changepoints. In order to achieve this separation, the following approach on estimating a distribution over recurrence of changepoint locations is proposed:

Changepoint probability vectors $p(c_t^{(n)}|\mathbf{x}_{1:t})$ as shown in Fig. 3.2 (third subplot) of N training signals are summed up across time steps $t = 1 \dots T$. For each training signal $n = 1 \dots N$, the cumulative probability mass $\sum_{n=1}^{N} p(c_t^{(n)}|\mathbf{x}_{1:t})$ increases at locations t of changepoints $c_t^{(n)}$ (i.e., locations t with non-zero probabilities $p(c_t^{(n)}|\mathbf{x}_{1:t})$) while staying the same at other time steps t where $p(c_t^{(n)}|\mathbf{x}_{1:t}) = 0$. Finally, dividing by N yields a distribution of empirically expected changepoint probabilities for each possible location t. The resulting distribution is depicted in the bottom subplot of Fig. 3.2 and henceforth referred to as *changepoint recurrence distribution (CPRD)*:

$$p\Big(c_t^{(1:N)}|c_t^{(n)}\Big) \triangleq \frac{\sum_{n=1}^{N} p\Big(c_t^{(n)}|\mathbf{x}_{1:t}\Big)}{N} \tag{3.5}$$

Recurrence of changepoints $c_t^{(n)}$ at locations t across signals $n = 1 \dots N$ is denoted by the term $c_t^{(1:N)}$. The distribution $p(c_t^{(1:N)}|c_t^{(n)})$ thus gives an empirical estimate how likely changepoints $c_t^{(n)}$ at locations t were present in all former N signals.

The CPRD allows incorporating further prior information. For instance, if time instants of processing step changes are available from the machine's control program, this deterministic prior knowledge can be utilized to complement the empirical information of observed changepoints. The information about changepoint recurrence incorporated in the CPRD can be utilized for a changepoint estimation more tailor-made to the given MHM data in two ways:

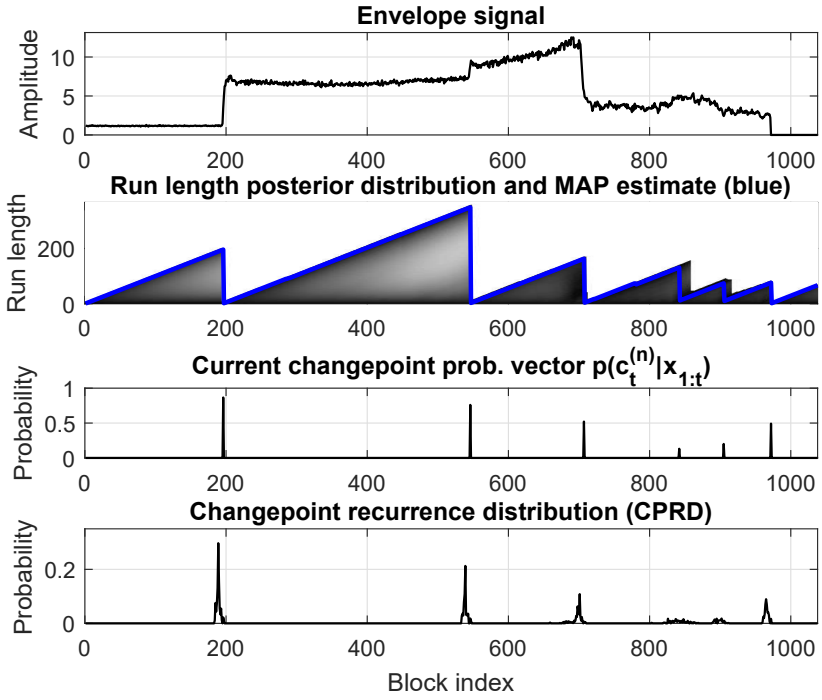


Figure 3.2: *CPRD estimation. Top: Exemplary envelope signal. Second: Pruned BOCPD solution. Run length log probabilities* $\log(p(r_t|\mathbf{x}_{1:t}))$ *depicted in gray, MAP estimates* $\hat{r}_t$ *as bold blue line. Third: Single changepoint probability vector* $p(c_t^{(n)}|\mathbf{x}_{1:t})$ *for signal from top. Bottom: Final CPRD estimate after accumulation and normalization of* N *training changepoint probability vectors.*

CPRD as Informative Hazard Function The CPRD can be used to replace the uninformative hazard function $H(\tau) = 1/\lambda$ introduced in [5]. This allows incorporating empirical information about the recurrence of observed changepoints directly into the changepoint prior $p(r_t|r_{t-1})$ and thus more robust estimations of recurrent signal segments in future signals by suppressing non-recurrent changepoints.

CPRD for Classification of BOCPD Changepoints An alternative approach is estimating all changepoints via BOCPD and using the CPRD to separate recurrent from non-recurrent changepoints in a subsequent step: By multiplying initial BOCPD changepoint estimates $p(c_t^{(n)}|\mathbf{x}_{1:t})$ with empirical CPRD probabilities $p(c_t^{(1:N)}|c_t^{(n)})$, a classification of changepoint estimates regarding their probability of being recurrent is obtained. This can be interpreted as applying Bayes' theorem:

$$p\left(c_t^{(n)}|c_t^{(1:N)}, \mathbf{x}_{1:t}\right) = \frac{p\left(c_t^{(1:N)}|c_t^{(n)}\right)p\left(c_t^{(n)}|\mathbf{x}_{1:t}\right)}{p\left(c_t^{(1:N)}\right)} \tag{3.6}$$

The CPRD $p(c_t^{(1:N)}|c_t^{(n)})$ acts as an estimate of the likelihood of changepoint recurrence. Initial BOCPD changepoint probabilities $p(c_t^{(n)}|\mathbf{x}_{1:t})$ are interpreted as prior estimates of recurrent changepoints for signal n. As the goal of the presented approach is finding non-zero probabilities $p(c_t^{(n)}|c_t^{(1:N)}, \mathbf{x}_{1:t})$, normalization to $p(c_t^{(1:N)})$ does not have to be considered:

$$p\left(c_t^{(n)}|c_t^{(1:N)}, \mathbf{x}_{1:t}\right) \propto p\left(c_t^{(1:N)}|c_t^{(n)}\right)p\left(c_t^{(n)}|\mathbf{x}_{1:t}\right) \tag{3.7}$$

Then, non-zero probabilities $p(c_t^{(n)}|c_t^{(1:N)}, \mathbf{x}_{1:t})$ can be used to indicate recurrent changepoints. Non-recurrent changepoints are then found as symmetric set difference between BOCPD changepoints $p(c_t^{(n)}|\mathbf{x}_{1:t})$ and recurrent changepoints.

For stationary behavior of normal changepoints, estimating the CPRD with a large number of normal training signals results in a smooth distribution. For a smaller number of training signals, postprocessing of the CPRD by fitting a kernel density estimator or GMM can similarly increase smoothness of the CPRD and thus yields more robust changepoint classification results. In the following sections, smoothing is performed by fitting a GMM, as this yields meaningful features (distance of changepoints to cluster centers, cluster membership probabilities, etc.) for a changepoint-related anomaly detection.

3.3 Experiments on Signal Segmentation

Disclaimer: Parts of this section were taken verbatim from own previous publications ([200] ©2018 IEEE, [202, 203] ©2019 IEEE).

This section first illustrates why cluster-based approaches are limited regarding their applicability to the given machine tool sensor data. Afterwards, cost and quality of signal segmentation via HMMs, the proposed GMM+FSM combination and BOCPD are compared. Finally, the benefit of signal segmentation via the CPRD extension of BOCPD is outlined for the predictive tasks of TCM and detection of sudden, process-related anomalies.

3.3.1 Data for Signal Segmentation

Signal segmentation results in this section are illustrated for five different data sets. Data sets DS_GM1, DS_GM2, DS_GM3 and DS_GM4 were recorded at a grinding machine, DS_TM data at a turning machine. All data sets were recorded with a vibration sensor attached to workpiece rest (DS_GM1,...,DS_GM4) or turning tool holder (DS_TM), respectively. Exemplary raw data records are depicted in Fig. 3.3. Most important characteristics of all data sets are summarized

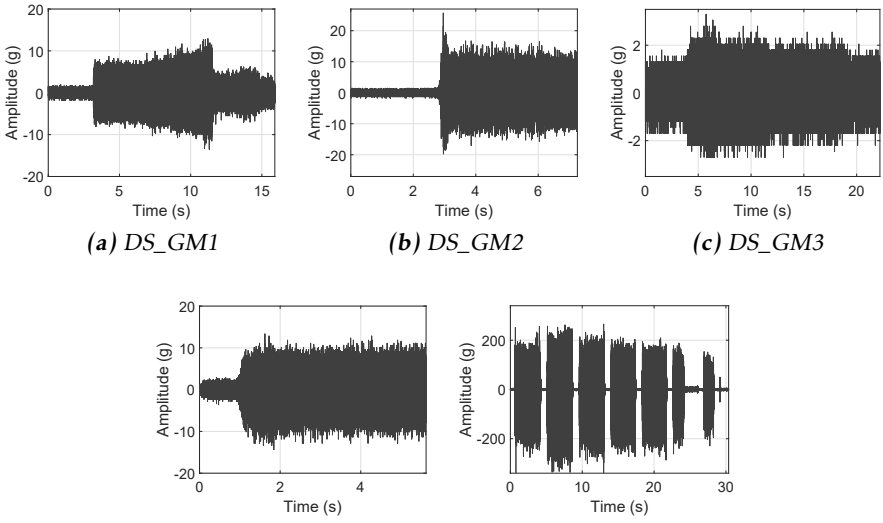


Figure 3.3: *Exemplary raw data records for various grinding machine data sets (DS_GM1 to DS_GM4) and a turning machine data set (DS_TM)*

in Table 3.1. Here, each data set consists of multiple data records. Each data record represents a sensor signal recorded during the machining of a single workpiece.

Data sets DS_GM1 and DS_GM2 consist of data recorded both for normal machining behavior as well as abnormal machining behavior. DS_GM1 data were recorded for the machining of a geometrically complex workpiece type. It consists of 312 normal sensor data records (recorded for a grinding wheel with normal behavior) and 118 data records for different degrees of severity of grinding wheel imbalance. DS_GM2 data are related to a less complex workpiece than DS_GM1. The DS_GM2 data set comprises 350 normal data records and 149 data records with machine part collisions. The collisions result in a single impulse-like artifact and thus one additional abnormal changepoint for abnormal DS_GM2 data records. DS_GM3 and DS_GM4 consist of only normal behavior data records. DS_GM3 data were included due to their small signal amplitudes, which are assumed challenging for signal segmentation methods due to the decreased signal-to-noise ratio (SNR) caused by coarse quantization.

DS_TM turning data illustrate the different segment structure for turning data compared to grinding data: Machining of workpieces with grinding machines is performed by a grinding wheel in a single stroke with different processing steps (roughing, finishing, etc.) and a workpiece kept still. Workpiece machining with turning machines is performed by rotating the workpiece (clamped between spindle chuck and tailstock) and gradually positioning the turning tool at the necessary positions relative to the workpiece during successive turning

Table 3.1: *Data sets and characteristics*

Data set	Source	Data records	Normal	Abnormal
DS_GM1	Grinding machine	430	312	118
DS_GM2	Grinding machine	499	350	149
DS_GM3	Grinding machine	97	97	-
DS_GM4	Grinding machine	1921	1921	-
DS_TM	Turning machine	66	66	-

strokes. The successive turning strokes can be identified easily in the depicted DS_TM signal and result in a highly different segment structure than that for grinding machine data sets.

3.3.2 Signal Segmentation by Clustering-based Methods

Features for Signal Segmentation

Clustering-based signal segmentation methods (GMM, HMM) rely on finding a sensible set of features where clusters of observations can be detected well. For the experiments in this thesis, power-related features proved to represent the information related to segment changes best. Suitable features are for example:

- Variance $\sigma^2 = \frac{1}{M-1} \sum_i (x_i - \bar{x})^2$
- Average rectified values $\mu_{abs} = \frac{1}{M} \sum_i |x_i|$
- RMS $RMS = \sqrt{\frac{1}{M} \sum_i x_i^2}$
- Frequency domain (FD) power $P_{FD} = \frac{1}{F} \sum_j |U_j|^2$

Here, x_i specifies the i-th raw signal sample and $|U_j|$ the j-th of $F = \frac{M}{2}$ spectral magnitudes. $\bar{x} = \frac{1}{M} \sum_i x_i$ is the average of samples x_i. Each time domain feature is extracted for M raw signal samples x_i. A fixed block length of $M = 1024$ raw signal samples proved to be suitable in the experiments. In the following section, the features μ_{abs} and P_{FD} are used for clustering-based signal segmentation. The features scores are normalized (z-scores).

GMM Model Selection

Apart from selecting a suitable set of features, the second challenge in clustering-based signal segmentation is selection of the optimal clustering model. In Fig. 3.4, exemplary GMM model selection results are depicted for the two features μ_{abs} and P_{FD} extracted from DS_GM3 signals. GMMs are trained for a number of cluster components $K = 1 \ldots 10$. AIC and BIC estimates are shown in the left part of Fig. 3.4. Assigning the optimal number of clusters in the knee point [287]

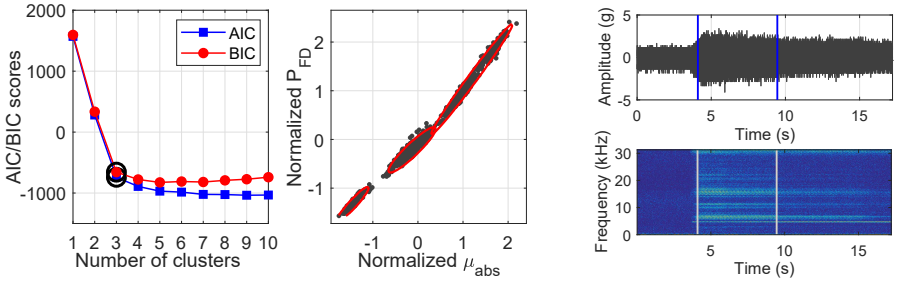


Figure 3.4: *GMM model selection results. Left: Best fit for AIC and BIC is found for three clusters (black circles). Middle: Scores for features μ_{abs} and P_{FD}. Equiprobability estimates for mixture component membership found with the best-fit GMM are plotted as contour lines. Right: Segment borders assigned for an exemplary signal (top: raw signal, bottom: spectrogram).*

of AIC and BIC plots results in an estimate of the best-fit model for three clusters. Segment borders assigned with this best-fit GMM match the differences in time domain (TD) envelope amplitude and FD energy distribution depicted in the right subfigure of Fig. 3.4.

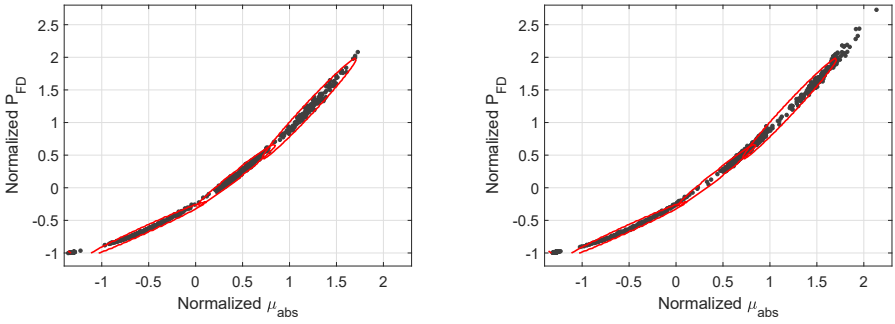
Drawbacks of Clustering-Based Signal Segmentation

Two main drawbacks were identified for clustering-based signal segmentation methods: The interdependency of suitable (power-related) features for signal segmentation and the dependency of these features on tool condition.

Both effects become apparent in Fig. 3.5. Here, two features μ_{abs} and P_{FD} extracted from successive blocks of two exemplary data records are depicted. The left subfigure shows feature scores for a data record related to a sharp grinding wheel and the right subfigure for a data record related to a dull grinding wheel. Feature scores in both subfigures are depicted for DS_GM1 data, as both drawbacks are most clearly identifiable in these data. For other data sets (e.g., DS_GM3 in Fig. 3.4), the effects are visually less obvious but effect the same problems as described in the following.

For both subfigures, feature scores cluster along a nonlinear function. The reason is the nonlinear dependency of both features μ_{abs} and P_{FD}. This dependency between the features leads to clusters collapsing along the nonlinear dependency function, which is disadvantageous for cluster identification by GMMs. Although decorrelation of features or relying on a single feature for cluster identification can mitigate this effect, the problem remains that no further relevant information is added by either of these adaptations.

A second major drawback of clustering-based signal segmentation is given by the drifting of scores for features extracted from successive data records. The drift of feature scores is due to worse tool condition: For dull grinding wheels, the signal power increases in certain frequency bands. This is outlined more detailed in later subsections. Thus, feature scores of successive data records continuously



***(a)** Sharp grinding wheel* ***(b)** Dull grinding wheel*

***Figure 3.5:** Scores of features μ_{abs} and P_{FD} extracted from two DS_GM1 records related to a sharp grinding wheel (left) and a dull grinding wheel (right)*

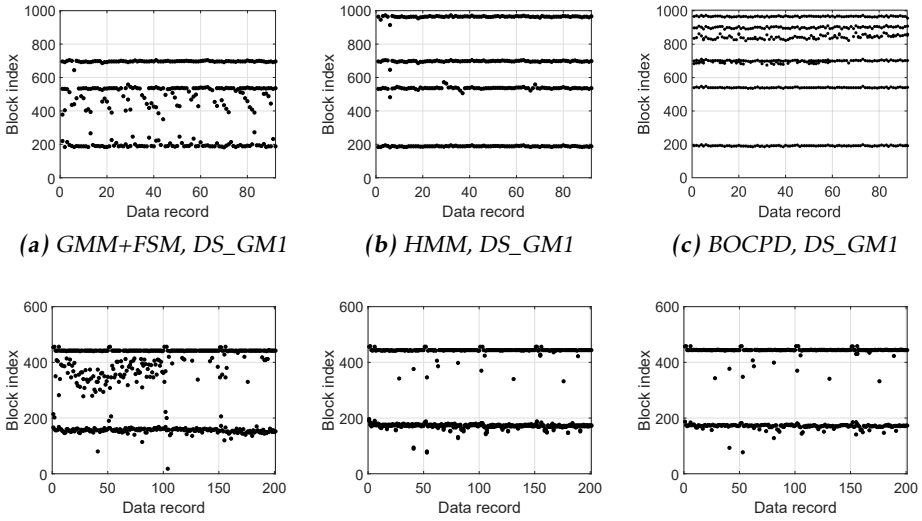
drift along the nonlinear dependency function between sharp and dull grinding wheels. Accordingly, the GMM clusters statically learned for the feature scores in Fig. 3.5a do not match the distribution of feature scores for a decreased tool condition in Fig. 3.5b anymore.

These effects of collapsing feature spaces as well as dependency of feature scores both on tool condition and signal segments illustrate why clustering methods relying on feature scores are not the optimal choice for signal segmentation. Although HMMs seem to be able to cope with both effects, resulting in a decent segmentation of signals, Bayesian changepoint detectors like BOCPD are a better match for the given sensor data. This is outlined in the following section.

3.3.3 Quality and Cost of Signal Segmentation

In this subsection, both quality and cost of signal segmentation via clustering-based approaches (GMM+FSM, HMM) and BOCPD are compared. The combination of GMM with FSM (GMM+FSM) is trained as described in Algorithm 1, the HMM model is selected via STACS as outlined in Section 2.2. Only the normal data subsets of data sets mentioned in Table 3.1 are considered for these comparisons. For a high-performant signal segmentation algorithm, segment borders are then assumed to be caused by processing step changes and thus expected to occur at similar block indices instances across all measurements. Non-recurrent segment borders are considered spurious. Additional recurrent segment borders (as observable for the BOCPD segment border estimates) are not considered spurious: The detailedness of signal segmentation (i.e., number of detected recurrent segments) is influenced by the choice of hyperparameters for the different segmentation approaches. Thus, detailedness of signal segmentation is not considered a quality measure but only recurrence of signal segments.

Fig. 3.6 illustrates a comparison of the quality of segmentation approaches



(a) GMM+FSM, DS_GM1 *(b) HMM, DS_GM1* *(c) BOCPD, DS_GM1*

(d) GMM+FSM, DS_GM2 *(e) HMM, DS_GM2* *(f) BOCPD, DS_GM2*

Figure 3.6: *Signal segment borders assigned by different algorithms for normal data of DS_GM1 (top row) and DS_GM2 (bottom row)*

(GMM+FSM, HMM and BOCPD). Segment border estimates are illustrated as black dots for all successive data records. Here, segment borders for BOCPD are assigned at changepoint estimates.

Across all data sets, BOCPD and HMM result in a more reliable segment border estimation than the GMM+FSM combination, i.e., segment border estimates occur at highly similar block indices in successive data records. In addition, the BOCPD approach can be elegantly extended to a computationally efficient discrimination of changepoints into recurrent (i.e., related to normal recurrent behavior of the repetitive processing step sequence) and non-recurrent (i.e., due to signal fluctuations or abnormal machining behavior) as will be discussed in the following section. This extension is not possible with the HMM-based approach as the HMM model does not adapt to alterations in the segment structure in a similarly implicit manner as the BOCPD model does: Abnormal changepoints would result in a decrease in predictive performance of signal segmentation, as the HMM model does not match the data and segment structure anymore. Thus, the main advantage of BOCPD is the fact that it comes without the necessity to explicitly relearn the segmentation model when the segment structure of the data changes (as opposed to clustering-based approaches).

Fig. 3.7 illustrates measurements of average computation time of the compared algorithms for segmentation of a single data record. The reported times are prediction times only, i.e., for already trained models. Time measurements were performed on an *Intel Core i7-6700* with 3.4 GHz without any optimization of MATLAB code or parallelization. HMMs shows a slight computational advan-

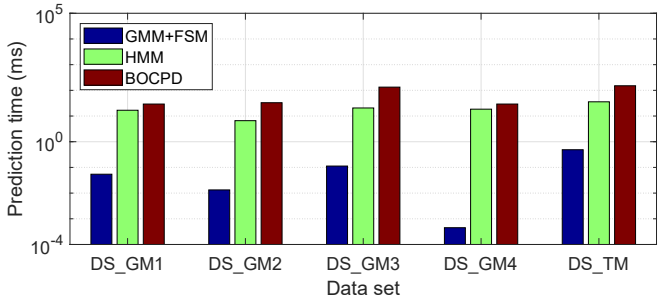


Figure 3.7: *Costs of signal segmentation by clustering-based and Bayesian changepoint detection methods. Time measurements performed in [172].*

tage compared to BOCPD. The GMM and FSM combination proposed in Subsection 3.2.1 illustrates a much smaller computational expense both than HMM and BOCPD but worse signal segmentation quality across all data sets (cf. Fig. 3.6).

3.3.4 Signal Segmentation by Bayesian Online Changepoint Detection and Extensions

In this subsection, the superiority of BOCPD to clustering-based signal segmentation is demonstrated in detail. The advantage is due to a combination of two phenomena:

1. BOCPD comes without the necessity of explicitly selecting an optimal model or to adapt the model to alterations in the segment structure like GMM- and HMM-based segmentation approaches. Thus, BOCPD yields reliable segmentation results even when the segment structure of the data changes. When aiming for a detection of similar and comparable segments even after alterations in the segment structure (due to additional, spurious changepoints), the CPRD extension to BOCPD introduced in Subsection 3.2.2 can be used. CPRD allows defining a more informative hazard function than the standard BOCPD hazard function in [5]. The resulting segmentation is more robust than using standard BOCPD and yields comparable segments. The benefit of comparable segments is outlined for the exemplary task of TCM in subsequent sections.

2. Thus, using the CPRD extension for defining an informative hazard function suppresses spurious changepoints. However, alterations in the signal segment structure causing these spurious changepoints are often a marker for abnormal machine behavior. Detecting these spurious changepoints thus allows for a computationally efficient detection of anomalies. Such a detection of spurious changepoints can be obtained based on the information of recurrent changepoint patterns represented by the CPRD. With an additional changepoint classification step after detecting changepoints

with standard BOCPD this CPRD information can be utilized in order to distinguish abnormal from normal changepoints. Exemplary features based on this ability to distinguish between normal and abnormal changepoints are presented, allowing for a low-cost but effective anomaly detection.

Fig. 3.8 illustrates these two approaches on utilizing the CPRD information: Figures 3.8a and 3.8b present CPRDs (black lines) estimated for DS_GM1 and DS_GM2. These CPRD estimates can be used as an informative hazard function in the BOCPD changepoint estimator as depicted in Figures 3.8c and 3.8d. Alternatively, the CPRDs can be utilized as a probability distribution to judge changepoints estimated via standard BOCPD regarding their probability of being recurrent. This is illustrated in Figures 3.8e and 3.8f.

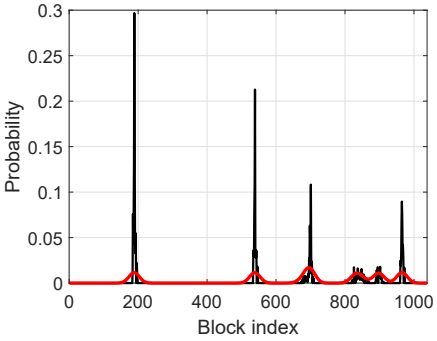
Fitting GMMs (red line) to CPRD estimates (black line) in Figures 3.8a and 3.8b results in a smoother probability distribution for the limited number of training examples as discussed in Subsection 3.2.2. Thus, for the experiments in the following sections, this GMM fit is used instead of the original CPRD estimate. Figures 3.8a and 3.8b visualizing the CPRD estimates are truncated in vertical direction for better visibility of the GMM fits.

CPRD as Informative Hazard Function

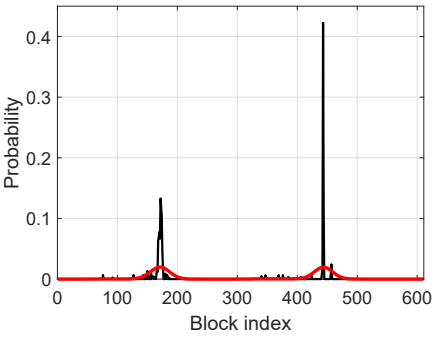
Results of utilizing these GMM fits as an informative hazard function in the BOCPD changepoint estimator are depicted in Fig. 3.8c. The top figure in Fig. 3.8c depicts an exemplary, abnormal DS_GM1 signal. The abnormal changepoint at block index 355 which is assigned when using the uninformative, standard BOCPD hazard function (middle figure of Fig. 3.8c) is suppressed by the informative hazard function (i.e., GMM fitted to CPRD) in the bottom figure of Fig. 3.8c. A similar behavior is illustrated in Fig. 3.8d for the abnormal changepoint at index 55 of an abnormal DS_GM2 signal. This confirms the validity of an informative CPRD hazard function for robust signal segmentation purposes (i.e., suppressing abnormal changepoints). Using the CPRD as more informative hazard function allows for a segmentation of signals only relying on recurrent changepoints, but does not allow the use as computationally efficient detector of sudden, process-related anomalies as discussed in the following section. Furthermore, the CPRD changepoint classification approach in the following section yields more robust estimates of recurrent changepoints. The reason for this is that, similar to the uninformative hazard function proposed in [5], likelihood and frequency of changepoints predicted with the CPRD hazard function depend on the amplitudes of the CPRD. Thus, a suitable factor for scaling CPRD amplitudes similar to the constant timescale parameter λ for the hazard function in [5] has to be found, either empirically or by hyperparameter optimization like in [209, 269].

CPRD for Classification of BOCPD Changepoints

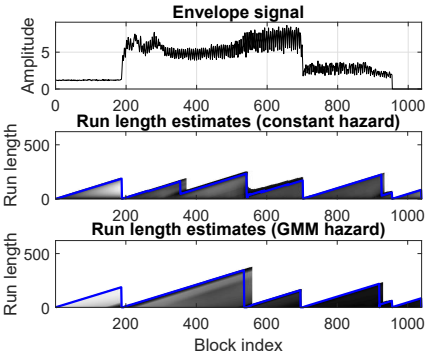
As an alternative to suppressing non-recurrent changepoints leveraging an informative hazard function as proposed in the previous section, recurrent changepoints can be separated from changepoints found via standard BOCPD. Results



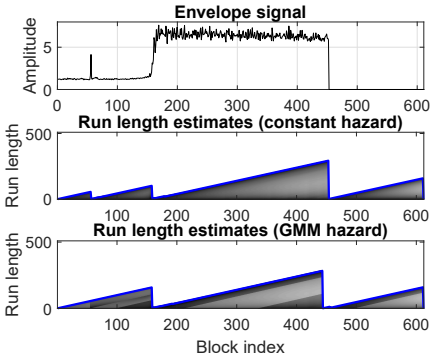
(a) CPRD, DS_GM1



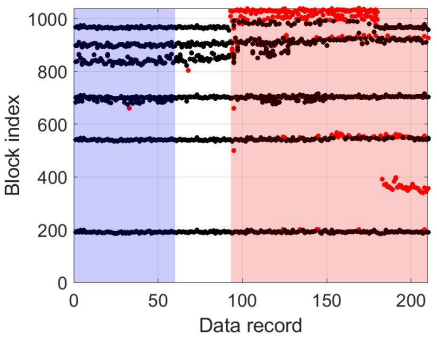
(b) CPRD, DS_GM2



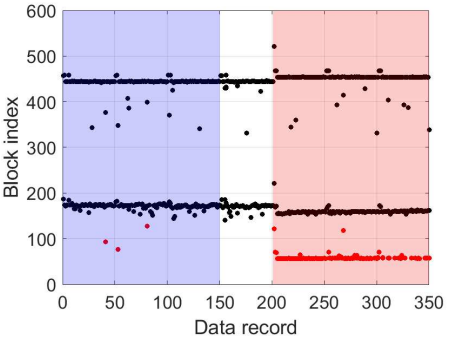
(c) CPRD hazard, DS_GM1



(d) CPRD hazard, DS_GM2



(e) CPRD classifier, DS_GM1



(f) CPRD classifier, DS_GM2

Figure 3.8: *(a), (b): CPRDs and GMM fits for data sets DS_GM1 and DS_GM2. (c), (d): Recurrent changepoint estimation with different hazard functions. Top: Abnormal envelope signal. Middle/bottom: Run length log probabilities (gray) and MAP estimates $\hat{r}_t$ (blue line) for standard BOCPD and GMM hazard functions. (e), (f): Classification of recurrent (black dots) and non-recurrent (red dots) changepoints. The blue shading illustrates the records used for estimating the CPRDs in the left column, the red shading marks data records consisting of abnormal signals.*

of classifying BOCPD changepoints with the GMM fitted to the DS_GM1 CPRD are depicted in Fig. 3.8e. Normal data consist of records nr. 1 to 91. Records nr. 1 to 60 (blue overlay) are used for estimation of the CPRD (Fig. 3.8a, black line). Recurrent changepoints likely under the GMM fitted to the CPRD are depicted as black dots. They allow dividing records into recurrent segments more robustly than via initial BOCPD changepoints (both black and red dots) similar to the approach in the previous section. For imbalanced grinding wheel anomalies (red overlay), additional non-recurrent changepoints occur between block indices 300 to 400 or block indices 970 to 1040.

For DS_GM2, abnormal machining (records nr. 201 to 350, red overlay) frequently effects machine part collisions, resulting in additional changepoints close to index 55 (red dots) (Fig. 3.8f).

These alterations in the changepoint pattern of DS_GM1 and DS_GM2, which were not detectable by the CPRD hazard approach in the previous section, support the benefit of being able to distinguish between recurrent and non-recurrent changepoints for a detection of sudden anomalies. At the end of this chapter, features including this knowledge of non-recurrent changepoints are proposed for the sake of detection of sudden anomalies.

3.3.5 Selected Predictive Tasks

As stated in the beginning of this chapter, signal segmentation is beneficial in MHM systems for the detection of both drifting as well as sudden anomalies. This is outlined in the following sections for the exemplary tasks of continuous TCM as well as detection of abnormal machine part contacts and signal form deterioration due to grinding wheel imbalances (sudden anomalies). For all predictive tasks illustrated in this subsection, data recorded at the workpiece support of grinding machines are used.

Tool Condition Monitoring

To maximize efficiency in machining sequences of modern workshops, processing time has to be minimized under the constraint of a specified minimum workpiece quality. The most important influencing factor for workpiece quality is the condition of the cutting tool, gradually getting dull by processing of workpieces. Thus, the cutting tool has to be resharpened (*dressed*). Being able to assess the necessity of dressing is thus beneficial for the sake of optimizing processing efficiency.

At present, tool condition is typically monitored via manual inspection by the machine operator. This is suboptimal, as manual inspection takes valuable time in which additional workpieces could be processed otherwise. In addition, manual inspection can be quite subjective and is thus error-prone. Automatic monitoring of process quality leads to a more stable quality of workpieces while freeing the operator from manual quality inspections.

As discussed throughout this chapter, machine tool data is characterized by its recurrent structure. Identifying these recurrent segments allows to extract features in parts of signals where they are related to each other. In addition

Algorithm 2: Learning of Discriminative Frequency Bands

Input: Multiple segmented data records $\boldsymbol{seg}_{dull}$, $\boldsymbol{seg}_{sharp}$
Output: Selected frequency bands $\boldsymbol{FB}_{sel}$

$\boldsymbol{Y} \leftarrow \text{INIT}(0)$ ▷ Initialize half spectrum $\boldsymbol{Y}$
for each pair $(seg_{i,dull}, seg_{i,sharp})$ **do**
  $\boldsymbol{X}_{i,dull} = \left|\text{FFT}(seg_{i,dull})\right|$
  $\boldsymbol{X}_{i,sharp} = \left|\text{FFT}(seg_{i,sharp})\right|$
  $\Delta\boldsymbol{X}_i = \boldsymbol{X}_{i,dull}(1 : M/2) - \boldsymbol{X}_{i,sharp}(1 : M/2)$ ▷ Addition assignment
  $\boldsymbol{Y}$ += $\Delta\boldsymbol{X}_i \odot \Delta\boldsymbol{X}_i$ ▷ Hadamard product
end for
$\bar{Y} = \frac{1}{M/2} \sum_{j=1}^{M/2} Y_j$ ▷ Average of half spectrum $\boldsymbol{Y}$
$\Delta\boldsymbol{Y} \leftarrow \text{SELECT}(\boldsymbol{Y} > \bar{Y})$
$\boldsymbol{FB}_{sel} \leftarrow \text{CONTIGREGIONS}(\Delta\boldsymbol{Y})$ ▷ Select contiguous regions of $\Delta\boldsymbol{Y}$
return $\boldsymbol{FB}_{sel}$

to finding comparable signal segments, automatically detecting most discriminative frequency bands between sharp and dull grinding wheels proved beneficial. So computing features in recurrent signal segments and in most discriminative frequency bands yields a tailor-made health indicator for tool condition. It can be calculated computationally efficient, nevertheless it describes the continuous degradation in tool condition well. Learning of frequency bands involves a unique additional computational burden but reduces running computational cost, as the health indicator is given by a feature of simple computational complexity and the monotonic trend of the health indicator allows to assess the grinding wheel condition via simple threshold-based classification.

Learning of discriminative frequency bands For signal segmentation, the CPRD extension of the BOCPD algorithm is used, which allows detecting recurrent segments seg_i across successive data records rec_i. The algorithm proposed for learning of discriminative frequency bands is summarized in Algorithm 2. The algorithm aims at identifying discriminative differences in distribution of spectral power for segments $seg_{i,dull}$ and $seg_{i,sharp}$ of data records rec_i related to dull and sharp grinding wheels. These discriminative differences can be found by computing the element-wise distance $\Delta\boldsymbol{X}_i$ of FFT magnitude spectra $\mathbf{X}_{i,dull}$ and $\mathbf{X}_{i,sharp}$ for these segments $seg_{i,dull}$ and $seg_{i,sharp}$. After element-wise squaring, discriminative spectra $\Delta\boldsymbol{X}_i$ for multiple pairs $(seg_{i,dull}, seg_{i,sharp})$ are accumulated. This is in order to compute a more robust estimate $\boldsymbol{Y}$ of discriminative differences in spectral power which generalizes across multiple pairs $(seg_{i,dull}, seg_{i,sharp})$. Most discriminative frequency bands $\mathbf{FB}_{sel}$ are then selected in contiguous regions above the average $\bar{Y}$ of the half spectrum $\boldsymbol{Y}$. Contiguous regions are selected with a minimum length of 1 % of the width of half spectrum $\boldsymbol{Y}$ (312.5 Hz), in order to avoid creating a meaningless multitude of scattered, narrow discriminative frequency bands $\mathbf{FB}_{sel}$. The resulting $\mathbf{FB}_{sel}$ can afterwards be used to filter

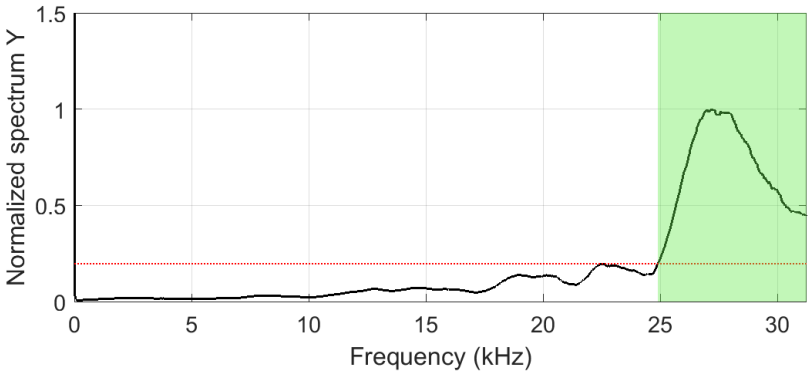


Figure 3.9: *Element-wise distance curve (solid black line) and selected frequency band (green patch) for DS_GM4. The frequency band is found relative to the half spectrum's average (dotted red line).*

the data records regarding most discriminative information for TCM.

Labels for sharp and dull grinding wheels are allocated via a sensor node attached to the dressing motor: For data recorded directly before dressing, the grinding wheel is assumed to be dull, while for data recorded directly after dressing, the grinding wheel is assumed to be sharp.

Resulting selected frequency bands $\boldsymbol{FB}_{sel}$ are illustrated in Fig. 3.9 exemplary for DS_GM4. For these data, only a single frequency band FB_{sel} is found, which is marked with a green patch. As outlined above, this frequency band FB_{sel} was found in contiguous regions of the normalized half spectrum $\boldsymbol{Y}$ (solid black line) exceeding the average of the half spectrum (dotted red line).

Online Processing: Extraction of TCM health indicator After segmentation of signals and determination of frequency bands $\boldsymbol{FB}_{sel}$, the feature $\mu_{abs} = \frac{1}{L}\sum_i |x_i|$ measuring the intensity of the signal samples in these comparable, recurrent signal segments and the most discriminative frequency bands $\boldsymbol{FB}_{sel}$ are extracted. For all data sets, the feature μ_{abs} is extracted across blocks of fixed length L at the beginning of the second of the found signal segments. The first segment in each signal involves the grinding wheel approaching the workpiece, where no contact between workpiece and grinding wheel exists (*air grinding*). The second segment is the one where most of the material removal takes place (*roughing*) and which evolved empirically as the segment where dulling of grinding wheels can be observed best.

The benefit of extracting the health indicator feature μ_{abs} only in comparable signal segments and most discriminative frequency bands $\boldsymbol{FB}_{sel}$ is visualized qualitatively in Fig. 3.10 (right subfigure): A monotonic feature score trend among examples of classes dull (red dots) and sharp (blue dots) evolves which resembles the continuous dulling of the grinding wheel. Also, labeled data points (red and blue dots) are separated more clearly compared to extracting μ_{abs} for the

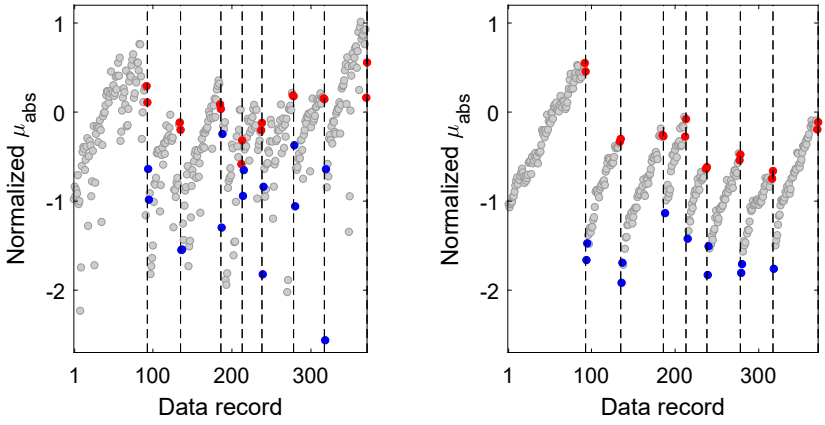


Figure 3.10: *Benefit of signal segmentation and learning of discriminative frequency bands for DS_GM4 data. Left: Normalized scores of feature μ_{abs} (gray) for complete data records and total frequency range. Dressing times are plotted as vertical dashed lines, feature scores for dull and sharp grinding wheels in red and blue, respectively. Right: μ_{abs} extracted in comparable, recurrent signal segments and discriminative frequency bands.*

complete spectral range and across records comprising all segments (left subfigure). Both effects in combination allow to define a simple threshold classifier in order to assess the point in time when dressing has to be applied. The threshold value can be chosen by the machine operator to steer the trade-off between optimal workpiece quality (early dressing) and long life time of the grinding wheel (late dressing), which in turn results in reducing the grinding cost per workpiece. The differences in terms of feature scores when dressing was applied (vertical dashed lines after red points) illustrate the non-optimality of choosing the point in time of dressing by visual inspection even for experienced machine operators: If dressing would have been applied always for the same tool condition, one would assume to observe highly similar features scores for the red points.

A quantitative evaluation of the benefit of preprocessing (i.e., signal segmentation and learning of frequency bands) is summarized in Table 3.2 for data sets DS_GM1, DS_GM3, DS_GM4 and DS_TM. DS_GM2 was omitted in the quantitative evaluation, as dressing was only applied two times in the complete data set and between both of these dressing events, multiple process adaptations were applied in order to make the machining process stable. This led to frequency band changes in between dressing events not related to dulling of the grinding wheel, which in turn led to inconsistencies in the feature score trends that are aimed to reveal and measure here.

Three performance measures are listed. Monotonicity of the feature score trend can be measured with the root mean squared error (RMSE) between feature scores and the fit of a trend function to these data. As evaluations revealed

Table 3.2: *Performance measures for TCM health indicator: Root mean squared error (RMSE), Cumulative squared distances (CSD), Trendability (Tre_2)*

	No preprocessing			With preprocessing		
Data set	**RMSE**	**CSD**	**Tre_2**	**RMSE**	**CSD**	**Tre_2**
DS_GM1	1.842e-2	1.041e-2	**0.9807**	**1.757e-2**	**5.578e-4**	0.9176
DS_GM3	0.3004	5.263e-3	0.4253	**0.1201**	**3.236e-4**	**0.8429**
DS_GM4	0.1635	2.727e-3	0.6476	**0.1537**	**4.710e-5**	**0.6752**
DS_TM	0.9449	0.0520	7.016e-2	**0.7652**	**0.0165**	**0.3449**

a restricted exponential growth of feature score trends (saturation effects due to restricted dulling of grinding wheel), feature scores are fit to the function $y = a - bc^{-dn+e}$ with n being the data record indices.

In addition, monotonicity can be measured by the cumulative squared distance (CSD) of feature scores: The squared differences between successive features scores are accumulated throughout a dressing cycle (i.e., between two dressing events). This favors on average small distances between adjacent feature scores and thus monotonic feature trends with a minimum amount of feature score outliers.

Finally, a trendability measure of feature scores is reported. This measure $Tre_2 \in [0, 1]$ is computed from the Spearman coefficient between the index of data records and related feature scores. The Spearman coefficient is a nonlinear correlation measure which identifies the long-term dependency of the feature scores and the time of their extraction, thus judging the sensibility of the feature scores regarding their capability to identify long-term trends. The Tre_2 measure is frequently applied in the TCM community [138].

In Table 3.2, smaller scores for measures RMSE and CSD and larger scores for measure Tre_2 represent enhanced monotonicity and trendability. Except Tre_2 for DS_GM1, extracting the feature μ_{abs} only in recurrent segments and frequency bands $\boldsymbol{FB}_{sel}$ (i.e., preprocessing) consistently results in better monotonicity and trendability of the computed tool condition health indicator. These quantitative results confirm the qualitative results depicted in Fig. 3.10.

Detection of Sudden Process Anomalies

Additional to benefiting the robustness of identifying recurrent segments for extraction of a TCM health indicator, changepoint recurrence estimation allows for an unsupervised detection of sudden process-related anomalies. This is outlined in the following section for two exemplary tasks: Detection of workpiece surface deteriorations (here due to imbalances in the grinding wheel, DS_GM1) and detection of abnormal machine part contacts (DS_GM2). For this, exemplary features are presented. These features build on the results for CPRD estimation and

Table 3.3: *F1 scores for changepoint-related features (©2019 IEEE [202])*

Data	Feature	N	AN1	AN2	AN3	AN4	F1 score
DS_GM1	N_{nc}	0	3	3	3	2	87.51 %
	D_c	13.12	29.41	30.04	33.38	36.86	97.14 %
	P_{ca}	42.23	99.99	99.99	99.99	100.0	94.29 %
	All						**99.05 %**
DS_GM2	N_{nc}	0	1				85.88 %
	D_c	3.78	30.63				84.80 %
	P_{ca}	1.76	99.63				83.93 %
	All						**97.86 %**

classification of BOCPD changepoints presented in Subsection 3.3.4.

The discrimination of BOCPD changepoints into recurrent and non-recurrent obtained by the changepoint classification approach in Subsection 3.3.4 can be used for an unsupervised detection of process-related anomalies. The following exemplary features are considered to be useful for a changepoint-related anomaly detection:

- N_{nc}: Number of non-recurrent changepoints in a signal.
- D_c: Average distance of BOCPD changepoints in a signal to closest cluster centers of the GMM fitted to CPRD estimates.
- P_{ca}: Maximum probability of all changepoints in a signal detected by BOCPD to be abnormal. This probability is calculated as follows: For all BOCPD changepoint locations t_c in this signal under review, calculate the complementary probability under the GMM fitted to the CPRD and take the maximum of these probabilities: $\max(1 - p_{GMM}(t_c))$.

The results of using these exemplary features for unsupervised anomaly detection are summarized in Table 3.3. Feature scores (columns N–AN4) are stated as medians of normal class (N) and abnormal (AN) classes. In DS_GM1, different degrees of grinding wheel imbalances yield multiple AN classes. F1 scores for anomaly detection with each feature are stated in the last column. F1 scores are computed as harmonic mean of precision and recall. An anomaly is classified for feature scores more than two normal class standard deviations distant from the normal class median.

Scores for single features show clear differences between normal and abnormal classes (columns N and AN) and result in a decent predictive quality as confirmed by the F1 scores. When the full feature set (i.e., a three-dimensional feature space) is considered, F1 scores improve to 99.05 % (DS_GM1) and 97.86 % (DS_GM2).

3.4 Conclusions

In this chapter, various methods for segmentation of sensor signals recorded from assorted types of machine tools were compared. Signal segmentation proved useful in order to find comparable segments in successive records for extraction of a tool condition health indicator feature. In addition, changes in the signal statistics can be used for a detection of sudden anomalies in the processing step sequence.

When comparing segmentation methods regarding quality and computational cost, the following findings can be stated:

- The proposed combination of GMMs with L2R FSMs yielded fast but unreliable signal segmentation results. Firstly, segment border estimates even during normal machining behavior frequently occurred at non-recurrent locations in data records, and can thus be considered spurious segment border estimates not representing the wanted processing step changes. Secondly, the challenge of finding a suitable set of features further complicates segmentation: Power-related features which proved most suited for segmentation are also correlated to a decreasing tool condition. This leads to a drift of test feature scores during each dressing cycle and consequently a progressive mismatch between the learned GMM-FSM segmentation model and the test data distribution.

- While HMM-based methods proved more robust to covariate drift than GMMs and were even on par with Bayesian Online Changepoint Detection (BOCPD) both regarding predictive quality and computation time, they can not adapt to long-term changes in the signal segment structure. This is due to the necessity to perform an explicit model selection step. Thus, when the segment structure changes, e.g., due to adjustments in the processing step sequence or changes of workpiece type, the trained HMM-based segmentation model does not fit the adapted signal statistics anymore.

- BOCPD on the other hand refrains from one-time learning a static segmentation model and is thus robust both to changes in the signal segment structure and the inherent drifting character of the sensor data, e.g., due to decreasing condition of the grinding wheel. In addition, BOCPD comes without the necessity to identify a suitable set of features.

Although BOCPD segmentation proved more robust to drifts and changes in the signal segment structure than GMMs and HMMs, standard BOCPD does not allow to explicitly model the cyclostationary behavior of the data-generating process and the resulting recurrent segment structure. When desiring to model this cyclostationary behavior, the introduced CPRD extension can be utilized in either of two ways: As an informative hazard function replacing the uniform hazard function in the standard BOCPD changepoint detector, or in an additional changepoint classification step after performing standard BOCPD changepoint detection. The latter approach yields a separation of changepoints into two subsets:

- Recurrent changepoints caused by the repetitive processing step sequence yield comparable signal segments for feature extraction. This proved useful for designing a novel tool condition health indicator. Extracting this health indicator only in recurrent signal segments and most relevant frequency bands increased scores for common monotonicity measures and trendability measures across evaluated data sets.
- Non-recurrent changepoints can indicate process-related anomalies, manifesting in a sudden alteration of the recurrent changepoint pattern. Designing features based on non-recurrent changepoints allows for an unsupervised detection of such anomalies. This was outlined for exemplary features and two predictive tasks, where F1 scores of 99.05 % and 97.86 % confirmed the sensibility of a changepoint-related detection of process anomalies.

3.5 Related Publications

C. Reich, C. Nicolaou, A. Mansour, and K. Van Laerhoven. Detection of machine tool anomalies from bayesian changepoint recurrence estimation. In *Proc. of the IEEE 17th International Conference on Industrial Informatics*, INDIN'19, pages 1297–1302, 2019.

C. Reich, C. Nicolaou, A. Mansour, and K. Van Laerhoven. Bayesian estimation of recurrent changepoints for signal segmentation and anomaly detection. In *Proc. of the IEEE 27th European Signal Processing Conference*, EUSIPCO'19, pages 1–5, 2019.

C. Reich, A. Mansour, and K. Van Laerhoven. Embedding intelligent features for vibration-based machine condition monitoring. In *Proc. of the IEEE 26th European Signal Processing Conference*, EUSIPCO'18, pages 371–375, 2018.

4

Modeling Non-Stationary Discrete Frequency Components

In Chapter 3, methods for encoding domain expertise concerning recurrent processing steps into the processing chain for designing tailor-made features were proposed. This allowed constructing meaningful health indicators for tool condition monitoring and for detection of sudden anomalies in the machining process. Tool condition monitoring is considered among most influential factors for quality of machined workpieces. Thus, the previous chapter focused mainly on *process monitoring*. This chapter will instead focus on modeling and understanding the spectro-temporal behavior of machine components in order to encode domain expertise in the process of feature engineering: During the measurement campaigns conducted for this thesis, different parts of machine tools were observed to be related to discrete frequency components (DFCs). Tracking of these non-stationary DFCs and subsequent assignment to machine parts allows for designing simple features custom-made for *condition monitoring* of these specific machine parts.

In order to identify the non-stationary behavior of DFCs during start-up of a grinding machine, a two-step approach is used:

1. First, peaks are identified in each frame of the time frequency distribution (TFD) computed for the sensor signals and the parameters amplitude and frequency related to these peaks are estimated. This task will be denoted in short as parameter estimation in this chapter.

2. Afterwards, the peaks are connected across successive frames in order to recover DFC tracks.

Both steps of parameter estimation and DFC tracking are explained in detail in Section 4.2. In Section 4.3, results on the quality of parameter estimation and DFC tracking for noisy data and the use of both techniques for selected predictive tasks are presented. As outcomes are known to be dependent on the amount of

noise in signals, the robustness of DFC tracking to noise (cf. Subsection 4.3.1) and its behavior for high noise levels are studied first. In order to be able to control the noise level, these experiments are performed for artificial signals mimicking the nature of the sensor data. In Subsection 4.3.2, results for parameter estimation and DFC tracking using the actual recorded sensor data are presented. First, parameter estimates and detected DFCs for these sensor data as well as their relation to certain machine parts are presented. Afterwards, techniques are proposed on leveraging the information about DFCs and their assignment to specific machine parts for the sake of exemplary condition monitoring tasks (mainly detection of imbalances in rotating machinery parts).

4.1 Motivation

Tracking of DFCs and assignment to related rotating machine components allows for creation of simple but expressive features for condition monitoring of these machine parts. This has been shown in previous studies for rotating parts like motors [213] and ball bearings [81]. Exemplary rotating parts in the focus of this thesis are grinding wheel, control wheel and dressing tool as well as the machined workpiece itself. For rotating machine parts related to machining of workpieces however, these discrete frequencies are typically concealed by the broadband and high-level energy distribution due to the machining of workpieces. When no workpieces are machined, the distribution of energy is more concentrated at the DFCs of interest. Such a non-machining phase occurs during start-up of the individual machine components, where the TF energy is clearly dominated by DFCs related to the machine parts of interest.

During start-up, these DFCs behave non-stationary, as rotating machine parts cannot directly be switched to full operational speed but are slowly accelerated until reaching the final velocity. Being able to track the non-stationary behavior of these DFCs and assigning them to related rotating parts is thus an important precedent step to extracting condition monitoring features as described above.

A sequential start-up of the machine parts would highly benefit this approach, as each DFC could be identified separately. Each DFC could then be assigned to the related rotating machine part currently started. For a typical machine tool as in focus of this chapter however, several rotating parts are started up in parallel, which makes the assignment task more difficult. In addition, spurious DFCs can further complicate the assignment of DFCs to rotating parts.

The individual identification of rotating parts despite of simultaneous start-up can be addressed by additional constraints on feasible velocities of rotating machine parts: Often, maximum permissible operational speeds are specified in the machine manual and can be used to upper-bound the search space for assignment of DFCs to potentially matching machine parts.

The challenge of spurious DFCs can be addressed by the capability of being able to track the non-stationary behavior of DFCs before becoming constant: Components being constant during the complete machine start-up are not related to rotating machine parts of interest and can thus be excluded from the assign-

Table 4.1: *Sequence of machine parts being switched on during start-up*

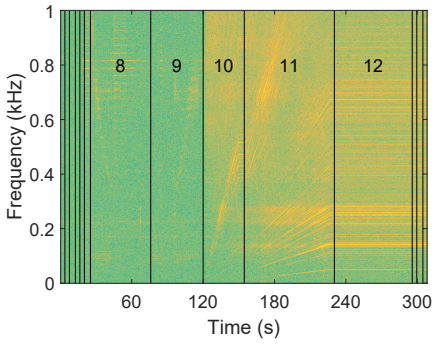
Index	Start-up step	Index	Start-up step
1	Return pump for cooling lubricant	8	Home drive
		9	Loading drive
2	Elec. main switch	10	Gr. wheel (no-load) $0 \rightarrow 900$ rpm
3	Pneumatics 1	11	Gr. wheel (no-load) $900 \rightarrow 3000$ rpm
4	Pneumatics 2	12	Warm-up drives
5	Feed pump for cooling lubricant	13	Handler and gripper start-up
		14	Handler slide-in linear
6	Display on	15	Handler feed air
7	Control voltage on		

ment of DFCs to potential matching rotating machine parts.

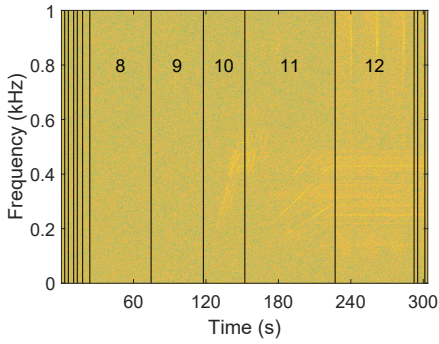
An exemplary start-up process of a grinding machine illustrating non-stationary DFCs is depicted in Fig. 4.1. TFDs for acceleration sensors and two sensor positions (workpiece support, grinding wheel housing) are illustrated. The TFDs in this chapter are spectrograms illustrated logarithmically in decibels. Numbers printed in the figure state the sequence of machine parts being switched on. The related machine parts and start-up steps are listed in Table 4.1. First, several components not related to rotating processes are started up (start-up steps 1 to 7), e.g., return pump and feed pump for cooling lubricants, electrical components and pneumatic components. During the home drive (start-up step 8), the compound slides carrying rotating machine parts are moved to the mechanical null point of the machine. Afterwards, the compound slides are driven in a manner simulating the default loading of workpieces for machining and back to the null point (start-up step 9). Then, the grinding wheel is accelerated to operational speed in two steps (start-up steps 10 and 11). No workpiece is loaded or machined during start-up of the grinding wheel, thus DFCs related to the workpiece cannot be tracked during start-up of the machine. The same is true for the dressing tool, which is started only during dressing cycles. After start-up of the grinding wheel, so-called warm-up drives are performed (start-up step 12), which simulate the movements of compound slides during standard machining behavior, but without grinding of workpieces. Finally, additional pneumatically operated parts related to the handlers and grippers are started up in steps 13 to 15. Handlers and grippers automatically position workpieces in the grinding gap for machining.

The positions and orientations of sensors used in this chapter are depicted in Fig. 4.2. Throughout this chapter, signals are evaluated for acceleration sensors only, which come with a better sensitivity than vibration sensors (cf. Table 1.1). In addition, all DFCs related to machine parts of interest are located in the acceleration sensors' bandwidth (1 kHz).

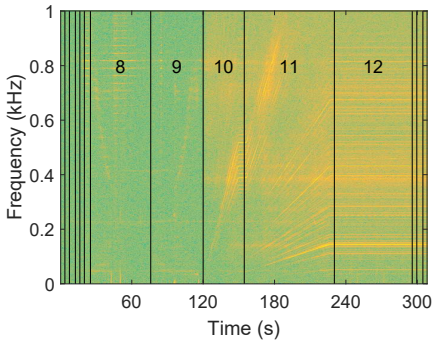
In general, detecting non-stationary DFCs and assigning them to machine parts is beneficial for two reasons:



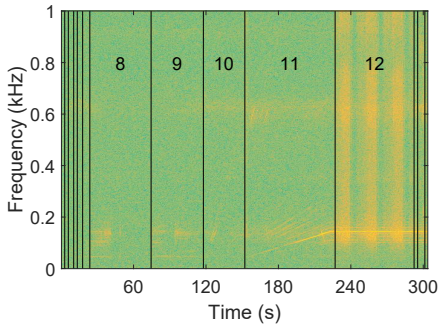
(a) *Acc. sensor, axis 1, gr. wheel housing*



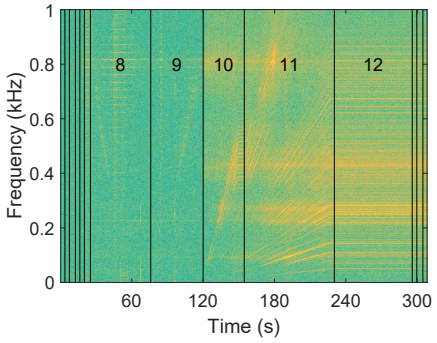
(b) *Acc. sensor, axis 1, workpiece support*



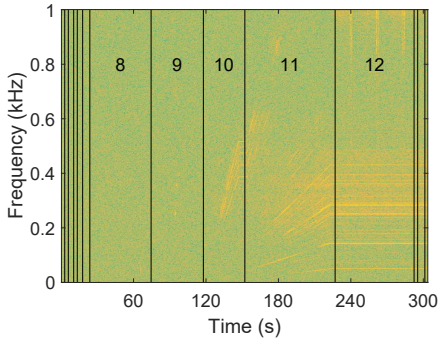
(c) *Acc. sensor, axis 2, gr. wheel housing*



(d) *Acc. sensor, axis 2, workpiece support*

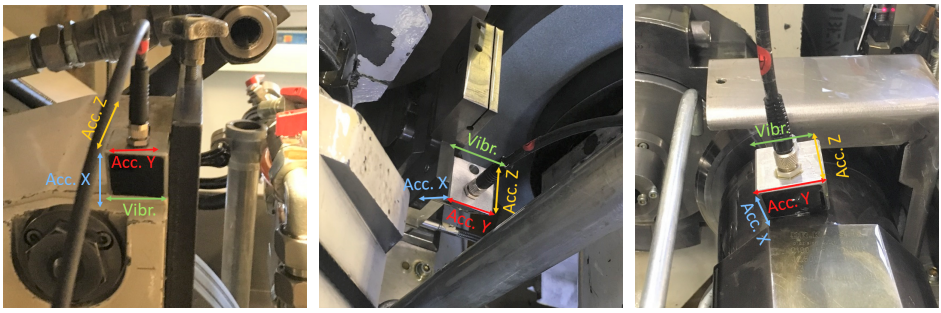


(e) *Acc. sensor, axis 3, gr. wheel housing*



(f) *Acc. sensor, axis 3, workpiece support*

Figure 4.1: *TFDs illustrating DFCs related to different machine parts during start-up of a grinding machine. Columns specify the sensor position (left: grinding wheel housing, right: workpiece support), rows the axes of acceleration sensors.*



(a) *Acceleration sensor, grinding wheel housing*

(b) *Acceleration sensor, workpiece support*

(c) *Acceleration sensor, dressing wheel motor*

Figure 4.2: *Positions and orientations of acceleration sensors used in this chapter*

- A semantic segmentation of the spectrograms depicted in Fig. 4.1 allows for an **understandable visualization of the current machine state** to machine operators: Tracked DFCs and related machine parts can be plotted as overlay on computed TFDs. This allows for a better understanding of the current machine behavior and earlier detection of machine or process anomalies.
- Tracking and assigning DFCs to specific rotating parts allows for a tailor-made **condition monitoring** of these machine parts. Here, machine parts condition monitoring is outlined for the specific tasks of imbalance detection in rotating machine parts like grinding wheels and dressing wheels. The methods discussed for imbalance detection are extendable to other rotating machinery like control wheels, when additional sensors are used (e.g., attached to the control wheel housing). In addition, the presented methods are expandable to process monitoring tasks. For example, tracking and assigning DFCs related to the rotational speed of the workpiece and its harmonics allow for the detection of workpieces with and insufficient roundness.

In the following section, estimation of parameters amplitude and frequency as well as tracking methods being the fundamental building blocks for approaching the two tasks in the previous list are presented.

4.2 Methods

As briefly discussed above, semantic segmentation of spectrograms computed during start-up of machine parts can be tackled by a multiple step approach. First, TFD peaks and their related parameters (amplitude and frequency) are estimated via non-stationary line spectrum estimation methods for each TFD

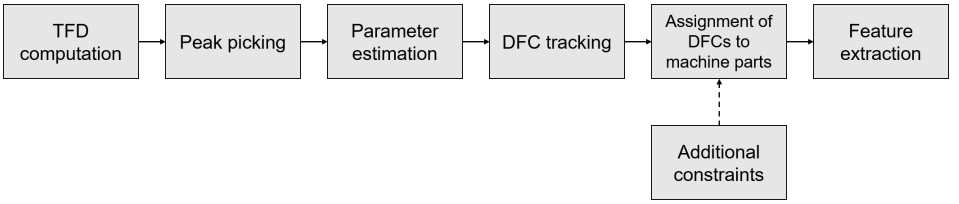


Figure 4.3: *Illustration of finding features for condition monitoring of specific machine parts employed in this chapter*

frame. Then, these successive peaks are connected across TFD frames by frequency tracking approaches in order to recover DFCs tracks. Afterwards, additional constraints regarding upper bounds of feasible rotational speeds for machine components are used in order to assign tracked DFCs to specific machine parts. This assignment step is best performed during start-up, due to the concentration of TF energy at the DFCs. Finally, assignment of DFC tracks to rotating machine parts allows to judge the health state of these machine parts with simple but effective features for the presented specific monitoring tasks. The approach is summarized in Fig. 4.3.

In the following section, methods applied for parameter estimation and DFC tracking in the upcoming experiments are outlined in detail. The derivation of both methods is in large parts inspired from the publications in [36] (for sinusoidal parameter estimation) and [171] (for DFC tracking) and reproduced here as a compressed version, for convenience of the reader. The derivations are outlined in time-continuous domains for reasons of generalizability and ease of explanations.

4.2.1 Estimation of Signal Model Parameters

Assume modeling each k-th frame of sensor signal $x(t)$ as a superposition of R_k generalized sinusoids $x_j(t)$, each of them representing a non-stationary DFC. Each generalized sinusoid $x_j(t)$ is represented by

$$x_j(t) = \exp\left(\sum_{i=0}^{Q} \alpha_{ij} m_i(t)\right) \tag{4.1}$$

with t being time, $m_i(t)$ being $Q+1$ basis functions and α_{ij} being complex weighting parameters. Choosing monomials $m_i(t) = t^i$ as basis functions allows approximating arbitrary functions with a Taylor expansion.

The parameters α_{ij} of this model can be effectively estimated via the distribution derivate method (DDM) [36]: As outlined in the following paragraphs, DDM allows to formulate a linear equation system for estimation of the complex parameters α_{ij} for elements $i = 1, \ldots, Q$ of the j-th generalized sinusoid.

For application of DDM, assume the component x_j being analyzed by a linear transform T_ψ (e.g., discrete Fourier transform (DFT) or discrete wavelet trans-

form (DWT)) with a set of basis functions $\psi(t) : \mathbb{R} \to \mathbb{C}$ and finite time domain support I_ψ (i.e., $\psi(t) = 0$ outside I_ψ) [36]:

$$T_\psi(x_j) = \langle x_j, \psi \rangle = \int_{-\infty}^{+\infty} x_j(t)\bar{\psi}(t)dt = \int_{I_\psi} x_j(t)\bar{\psi}(t)dt \tag{4.2}$$

Here, $\bar{\psi}$ denotes complex conjugates of basis functions ψ and $\langle \cdot, \cdot \rangle$ the inner product operator. The basis functions ψ are assumed to be normalized (i.e., $T_\psi(\psi) = \langle \psi, \psi \rangle = 1$) and being part of the set of all continuously differentiable functions C^1 on the interval I_ψ. Then, using integration-by-parts on the inner product related to the linear transform in Eq. 4.2, one obtains

$$\left[x_j(t)\bar{\psi}(t)\right]_{-\infty}^{\infty} = \langle x_j'(t), \psi(t) \rangle + \langle x_j(t), \psi'(t) \rangle, \tag{4.3}$$

where the apostrophe $(\cdot)'$ denotes differentiation. For finite time domain support I_ψ one has $\lim_{t \to \pm\infty} \bar{\psi} = 0$, thus the left-hand side equals zero and the distribution derivate rule is obtained:

$$\langle x_j'(t), \psi(t) \rangle = -\langle x_j(t), \psi'(t) \rangle \tag{4.4}$$

Substituting Eq. 4.1 into Eq. 4.2 and using $m_i(t) = t^i$, Eq. 4.4 is rewritten as

$$\sum_{i=1}^{Q} \alpha_{ij} \int_{I_\psi} m_i'(t)x_j(t)\bar{\psi}(t)dt = -\int_{I_\psi} x_j(t)\bar{\psi}'(t)dt \tag{4.5}$$

Using the definition of T_ψ in Eq. 4.2, one can rewrite the equation as

$$\sum_{i=1}^{Q} \alpha_{ij} T_\psi(x_j \cdot m_i') = -T_{\psi'}(x_j), \tag{4.6}$$

or, expanded to an equation system linear in the complex parameters α_{ij}, as:

$$\underbrace{\begin{pmatrix} T_{\psi_1}(x_j m_1') & \cdots & T_{\psi_1}(x_j m_Q') \\ \vdots & \ddots & \vdots \\ T_{\psi_R}(x_j m_1') & \cdots & T_{\psi_R}(x_j m_Q') \end{pmatrix}}_{\mathbf{A}} \cdot \underbrace{\begin{pmatrix} \alpha_{1j} \\ \vdots \\ \alpha_{Qj} \end{pmatrix}}_{\boldsymbol{\alpha}_j} = \underbrace{\begin{pmatrix} -T_{\psi_1'}(x_j) \\ \vdots \\ -T_{\psi_R'}(x_j) \end{pmatrix}}_{\mathbf{b}} \tag{4.7}$$

Again, $\psi_1, \ldots, \psi_R$ are the basis functions of transform T_ψ, $\mathbf{A}$ is an $R \times Q$ matrix, $\boldsymbol{\alpha}_j$ a vector of length Q and $\mathbf{b}$ a vector of length R. Then, $\boldsymbol{\alpha}_j$ has a single unique solution

$$\boldsymbol{\alpha}_j = \mathbf{A}^+ b = (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H b \tag{4.8}$$

where $\mathbf{A}^+$ denotes the Moore-Penrose pseudoinverse of matrix $\mathbf{A}$.

Thus, by computing $T_\psi(x_j m_i')$ for elements $i = 1,\dots,Q$ and $T_{\psi'}(x_j)$ for all derivatives of basis functions $\psi_1',\dots,\psi_R'$, the complex coefficients α_{ij} can be recovered efficiently. Note, however, that due to the differentiation of x_j in Eq. 4.4, the parameter α_{0j} vanished from the sum in Eq. 4.1. This parameter has to be estimated separately after the parameters α_{ij} for $i = 1,\dots,Q$ have been obtained via Eq. 4.8. Betser proposes two approximate least square estimators, of which the first alternative relying on the basis function $\psi_{\hat{r}}$ covering the largest energy proportion of x_j is applied in the following sections [36]:

$$\hat{r} = \arg\max_r |T_{\psi_r}(x_j)| \tag{4.9}$$

$$\hat{\alpha}_{0j} = \log(T_{\psi_{\hat{r}}}(x_j)) - \log(T_{\psi_r}(\gamma)) \tag{4.10}$$

Here, γ is defined as the signal parts not relying on α_{0j}:

$$x_j(t) = e^{\alpha_{0j}}\gamma(t) \tag{4.11}$$

The instantaneous log-amplitude, phase and normalized angular frequency of each sinusoid j can be obtained from the complex coefficients α_{ij} via

$$a_j^{(k)}(t) = \mathfrak{Re}\left(\sum_{i=0}^{Q}\alpha_{ij}t^i\right) \tag{4.12}$$

$$\phi_j^{(k)}(t) = \mathfrak{Im}\left(\sum_{i=0}^{Q}\alpha_{ij}t^i\right) \tag{4.13}$$

$$f_j^{(k)}(t) = \frac{f_s}{2\pi}\frac{d\phi_j^{(k)}(t)}{dt} = \frac{f_s}{2\pi}\mathfrak{Im}\left(\sum_{i=0}^{Q}\alpha_{ij}it^{i-1}\right) \tag{4.14}$$

like specified in [171]. Here and in the following equations, the superscript (k) is used to denote parameter estimates in frame k. The parameter estimates $a_j^{(k)}$ and $f_j^{(k)}$ estimated for each frame k are used in the successive step of DFC tracking.

4.2.2 Discrete Frequency Component (DFC) Tracking

For DFC tracking, the signal model $x(t)$ defined in Eq. 2.14 modeling the signal as a superposition of R_k generalized sinusoids per each frame k of an STFT spectrogram of the sensor signal $x(t)$ is reconsidered [171]:

$$x(t) \approx \sum_{j=1}^{R_k} x_j(t) = \sum_{j=1}^{R_k}\exp\left(\sum_{i=0}^{Q}\alpha_{ij}t^i\right), \tag{4.15}$$

where i is the degree of monomial t^i, α_{ij} are the complex parameters of generalized sinusoid j and R_k the number of generalized sinusoids in frame k. DFC

tracking can then be approached by connecting best-matching TFD peaks of each frame k found during the parameter estimation step. Matching criteria are typically based on continuity of parameter estimates found in the previous section for the complex parameters α_{ij}. As discussed in Subsection 2.3.2, tracking methods formulate matching criteria based on HMMs [119], Markov renewal processes [243] or linear programming [73, 171] among others. In this thesis the linear optimization approach presented in [171] is applied, as the approach is computationally efficient and considered robust to noise in sensor signals.

For the assignment of TFD peaks u in frame $k-1$ to peaks v in frame k, Neri et al. [171] specify multivariate Gaussian cost functions

$$A_{uv} = 1 - \exp\left(-\left(\frac{\Delta a_{uv}^2}{2\sigma_a^2} + \frac{\Delta f_{uv}^2}{2\sigma_f^2} \right) \right) \tag{4.16}$$

for useful assignments and

$$B_{uv} = 1 - (1-\delta)A_{uv} \tag{4.17}$$

for spurious assignments. Standard deviations σ_a and σ_f as stated in Eq. 4.16 are defined by

$$\sigma_a^2 = \zeta_a^2 / (2\ln(\delta - 2) - 2\ln(\delta - 1)) \tag{4.18}$$

$$\sigma_f^2 = \zeta_f^2 / (2\ln(\delta - 2) - 2\ln(\delta - 1)) \tag{4.19}$$

The parameters δ, ζ_a and ζ_f determine the points of transition between useful and spurious assignments as outlined in the following paragraphs.

Useful assignments are defined as those assignments which satisfy continuity constraints of parameter values $a_j^{(k)}(t)$ and $f_j^{(k)}(t)$ across successive frames k [171]. The continuity across frames is measured by amplitude assignment gaps Δa_{uv} and frequency assignment gaps Δf_{uv}. For computation of Δa_{uv} and Δf_{uv}, parameter values $a_j^{(k)}(t)$ and $f_j^{(k)}(t)$ always at time indices $t = 0$ located at the center of frames k are considered. The dependency on t can thus be dropped in all following equations.

Assignment gaps Δa_{uv} and Δf_{uv} are defined as

$$\Delta a_{uv} = a_u^{k-1}(H/2) - a_v^k(-H/2) \tag{4.20}$$

$$\Delta f_{uv} = f_u^{k-1}(H/2) - f_v^k(-H/2) \tag{4.21}$$

Amplitude estimates a_u^k and frequency estimates f_u^k per peak u are computed as presented in Equations 4.12 and 4.14, respectively. H denotes a so-called hop size such that time steps of parameter estimation are $t_k = kH/f_s$, with f_s being the sampling frequency. The hop size H in turn is defined by N/H_f, where N is the number of signal samples per TFD frame and H_f a so-called hop factor. For $H_f \neq 1$, successive TFD frames overlap, which allows estimating parameters at smaller time distances than specified by the inverse of frame rate $1/N$.

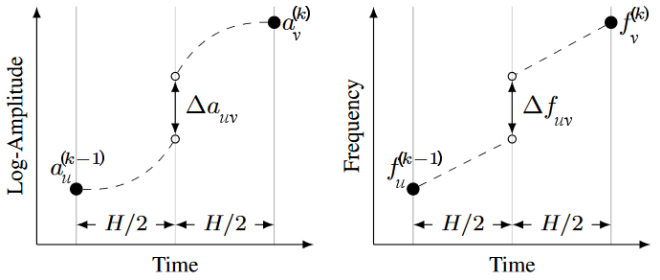


Figure 4.4: *Illustration of peak connection like proposed in [171] for a polynomial order $Q = 2$. Small assignment gaps for amplitudes Δa_{uv} (Eq. 4.20) and frequencies Δf_{uv} (Eq. 4.21) make connection of related peaks more likely. Figure adapted from [171] with kind permission of the authors.*

A visual interpretation of the meaning of amplitude and frequency gaps is sketched in Fig. 4.4 for a polynomial order of $Q = 2$ (i.e., considering monomials up to a degree of 2). Due to the at most linearly evolving nature of DFCs in our data, a polynomial order of $Q = 2$ is sufficient: $Q = 2$ assumes linear connections of per-frame frequency parameter estimates between TFD frames and a superposition of linear and quadratic connections between per-frame amplitude estimates (cf. Equations 4.12 and 4.14). Furthermore, for $Q = 2$, the DDM-based parameter estimation described in Subsection 4.2.1 is related to so-called reassigned spectrograms as outlined in [36]. Reassigned spectrograms exhibit desired characteristics for the estimation of DFCs: They can be applied to compensate for the spreading of TF energy (caused by STFT computation) by reassigning the energy to the true frequency values. Reassignment methods were pioneered by Kodera et al. in [127] for spectrograms and generalized in [77] to other bilinear TFDs.

Useful matrix elements A_{uv} and spurious matrix elements B_{uv} are combined in a single cost matrix with elements C_{uv} which will ultimately be considered during optimization of the peak assignment problem. Elements C_{uv} are obtained as follows:

$$C_{uv} = \min\{A_{uv}, B_{uv}\} \tag{4.22}$$

Both Eq. 4.22 and the influence of parameters δ, ζ_a and ζ_f on the transition between useful and spurious assignments are illustrated in Fig. 4.5.

Optimal assignments are finally found via the following linear programming problem:

$$\min \quad \sum_{u=1}^{R} \sum_{v=1}^{R} C_{uv} X_{uv} \tag{4.23}$$

$$\text{subject to} \quad \sum_{u=1}^{R} X_{uv} = 1 \quad v = 1 \ldots R \quad \text{and} \quad \sum_{v=1}^{R} X_{uv} = 1 \quad u = 1 \ldots R \tag{4.24}$$

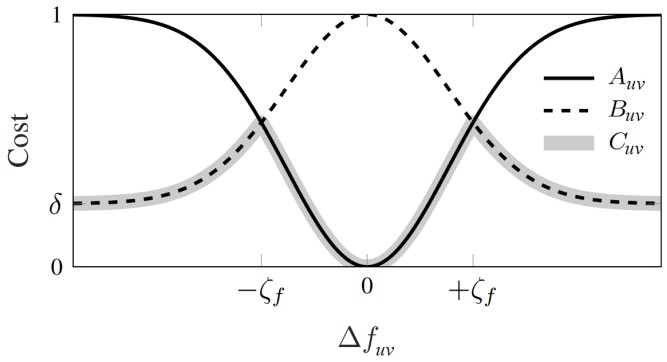


Figure 4.5: *Illustration of the influence of parameters δ and ζ_f on the points of transition between useful cost A_{uv} and spurious cost B_{uv} in the combined cost C_{uv}. Figure adapted from [171] with kind permission of the authors.*

which can be solved in polynomial time by the Hungarian algorithm [130]. Here, X_{uv} are binary variables that equal 1 if peak u is assigned to peak v and 0 otherwise [171]. $R = \max\{R_{k-1}, R_k\}$ is the largest of the two numbers of peaks R_{k-1} and R_k.

4.3 Results

In this section, experiments for designing tailor-made features for condition monitoring of rotating machine parts are presented. The performance of the parameter estimation and DFC tracking methods described in the previous section are illustrated for artificial data (Subsection 4.3.1) and data measured with sensors attached to a grinding machine (Subsection 4.3.2). For the experiments producing these results, Neri et al.'s implementation of these parameter estimation and DFC tracking methods was adapted to the nature of the given sensor data. The original implementation written for their publication [171] can be accessed via [170]. Afterwards, features for monitoring of rotating machine parts are presented.

As outlined in [171], the success and quality of DFC tracking is highly dependent on the amount of noise present in the computed TFDs being used as basis for parameter estimation. For high amounts of noise, detected tracks illustrate random directions, whereas TFD regions being dominated by signal components expose aligned tracks. The **first major question** in this chapter is thus, whether the high amount of noise present in the spectrograms for the recorded sensor data (cf. Fig. 4.1) hinders a proper recovery of DFCs. Thus, in Subsection 4.3.1, the effect of an increasing amount of noise onto the performance of parameter estimation and DFC tracking is studied. For this, the RMS value of additive white Gaussian noise (AWGN) added to artificial signals mimicking the recorded sensor data is varied.

The **second major question** is which machine parts can be identified from spectrograms and DFC tracks. In Subsection 4.3.2, this is discussed for the start-

up process of an exemplary grinding machine. The grinding wheel rotational speed can be distinguished from other DFCs by leveraging upper bounds on feasible rotational velocities. A similar approach for detecting imbalances in the dressing wheel is outlined.

Finally, the **third major question** is how to use the knowledge of rotating parts and assigned DFCs for the sake of a tailor-made feature extraction. An approach for the task of detecting grinding wheel imbalance is proposed and discussed in detail, similar approaches for the tasks of detecting dressing wheel imbalance and detection of workpieces with insufficient roundness are outlined.

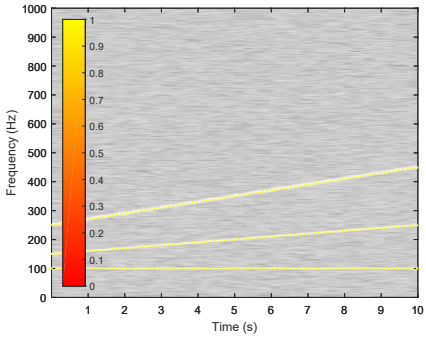
4.3.1 Noise Variations for Artificial Data

This subsection covers the first of the major questions formulated in the former subsection, the robustness of DFC tracking to high amounts of noise in the sensor data. In [171], Neri et al. demonstrate the robustness of their DFC tracking approach to noise by adding AWGN to artificial data. Their findings are of interest for this thesis as the sensor data presented here illustrate high noise levels especially during start-up of the grinding machine (cf. Fig. 4.1).

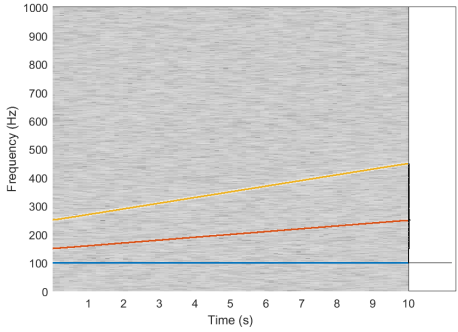
In this subsection, the tracking methods of Neri et al. are validated on artificial data mimicking the nature of the recorded sensor data. During start-up, the frequency of each discrete component is restricted to be either constant or to evolve linearly (cf. Fig. 4.1). After run up to operational speed, the DFCs related to started rotating machine parts become constant. The robustness to AWGN is thus validated on an artificial test signal constructed from a superposition of equally weighted constant DFCs and linear chirp signals (i.e., showing linearly evolving frequencies).

In Figure 4.6, parameter estimates (left column) and tracked DFCs (right column) are presented for this test signal with different SNR values. Amplitude estimates a_j^k in the left column are visualized in yellowish and reddish colors as values normalized to the maximum amplitude estimate encountered, thus mapping them to the range $[0, 1]$. Amplitude estimates of TFD peaks are shown as overlay plot to noisy spectrograms (visualized in gray color scaling). As the chirps constituting the artificial test signal have constant amplitude, all peaks illustrate the same yellowish color. For DFC tracking (right column), marginal histograms of tracked DFCs are additionally plotted. They are created by accumulating DFC tracks over time. Thus, peaks of the histograms occur in frequency bins where tracks are most often observed. Leveraging marginal histograms proved useful for tracking of stationary DFCs in noisy sensor data as outlined in the following paragraphs for artificial data described above and in Section 4.3.2 for measured sensor data.

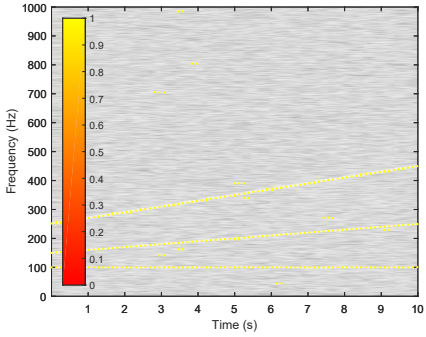
For an SNR of 15 dB (top row), the signal energy clearly dominates the TF plane as verified by spectrograms in both subfigures. For higher amounts of noise (SNR = 5 dB, middle row), the same DFC tracks are found as for an SNR of 15 dB, but DFC tracks start to illustrate discontinuities. For even higher amounts of noise (SNR = 0 dB, bottom row), DFC tracks become even more fractionated. In addition, one observes randomly directed DFC tracks constructed between



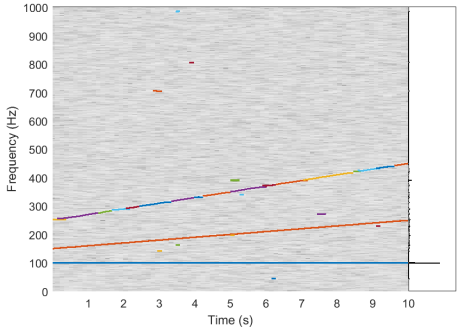
(a) Parameter estimates, SNR = 15 dB



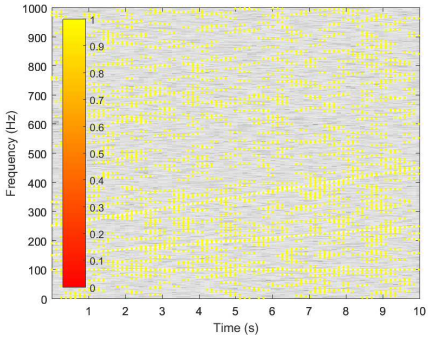
(b) Frequency component tracks, SNR = 15 dB



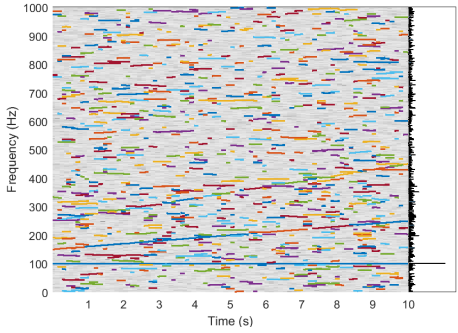
(c) Parameter estimates, SNR = 5 dB



(d) Frequency component tracks, SNR = 5 dB



(e) Parameter estimates, SNR = 0 dB



(f) Frequency component tracks, SNR = 0 dB

Figure 4.6: *Results of parameter estimation for detected peaks (left column) as well as tracked DFCs and related marginal histograms (right column) for an artificial superposition of order 2 generalized sinusoids (i.e., linear chirps) and different SNRs. Higher noise levels result in more fractioned frequency component tracks. In the marginal histograms however, peaks for constant frequencies can still be identified reliably.*

spurious peaks caused by the high noise. This is in accordance with the results presented in [171]: High amounts of noise make a continuous tracking of DFCs challenging.

In the right column of Fig. 4.6, the marginal histograms of DFC track values are plotted. Even for high amounts of noise (SNR = 0 dB, right column), the marginal histogram allows to robustly detect the constant DFC at 100 Hz. This fact will be used in the following section during the experiments with real-world data.

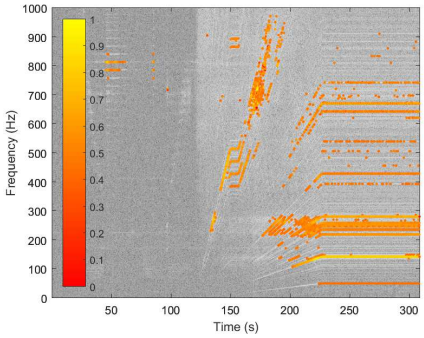
4.3.2 DFC Tracking and Assignment for Measured Sensor Data

In this subsection, the second major question formulated in the beginning of Section 4.3 is considered: The set of machine parts which can be identified from detected DFC tracks. For this, results for parameter estimation and DFC tracking during start-up of machine tools and other machine states (sharpening of grinding wheels, machining of workpieces) are presented. For parameter estimation, TFD peaks are picked by a heuristically found magnitude threshold like in [170, 171].

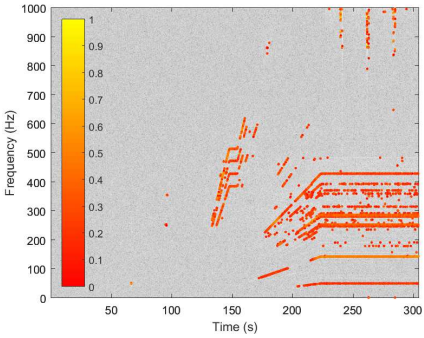
Figure 4.7 illustrates parameter estimates (top row) and tracked DFCs (bottom row) for signals recorded with acceleration sensors attached to grinding wheel housing (left column) and workpiece support (right column) of a grinding machine during start-up. TFDs for both sensors illustrate multiple DFCs starting to occur with the start-up of the grinding wheel around second 120. Some of these components become temporarily stationary around second 150 occupying a frequency band between 350 and 550 Hz. This is a point in time between start-up steps 10 and 11, where the grinding wheel is accelerated first from 0 to 900 rpm and then from 900 to 2400 rpm, respectively (cf. Table 4.1 and Figures 4.1b to 4.1f). These DFCs are thus related to machine components involved with the control chain of the grinding wheel (i.e., grinding wheel, motor, etc.). The signals behave stationary beginning with start-up step 12 at second 220 (warm-up drives, cf. Table 4.1). As all of these machine parts are started at the same point in time, they can not be distinguished without leveraging additional information.

Additional information for assignment of machine parts to detected DFCs can be defined by regions on feasible operational speeds of machine parts. This is exemplary outlined in detail for grinding wheels. For the grinding machine being the source of sensor data presented in this chapter, grinding wheel rotational speeds are upper-bound by a value of 7200 rpm which corresponds to 120 Hz [216]. The region of permissible grinding wheel rotational speeds is marked by a green patch in the marginal histograms in Figures 4.7c and 4.7d. One can observe, that the 50 Hz component is the only detected DFC in this region which can thus be assigned to the grinding wheel rotational speed. Furthermore, the 50 Hz component is not constantly present but evolves linearly before reaching constancy at 50 Hz. The 50 Hz component can thus safely be excluded from being a spurious (e.g., power line frequency) component.

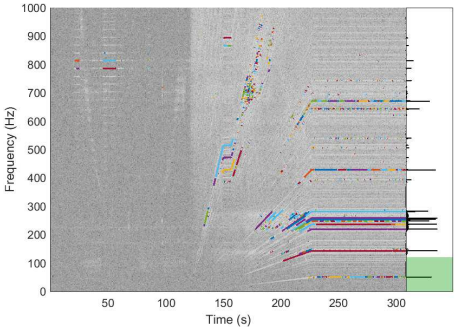
For both acceleration sensors, this 50 Hz component can be detected in the marginal histograms (cf. Figures 4.7c and 4.7d), although the related DFC track



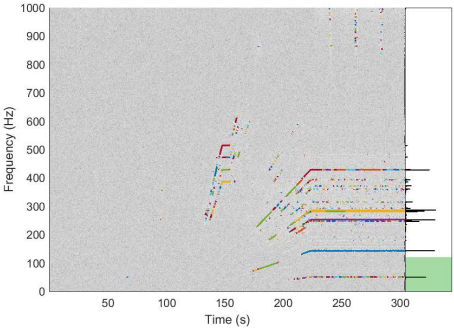
(a) Parameter estimates, sensor at grinding wheel housing



(b) Parameter estimates, sensor at workpiece support



(c) Frequency component tracks, sensor at grinding wheel housing



(d) Frequency component tracks, sensor at workpiece support

Figure 4.7: *Results of parameter estimation for detected peaks (top row) as well as tracked DFCs and related marginal histograms (bottom row) for the complete start-up of a grinding machine. The left column depicts a signal from the acceleration sensor attached to the grinding wheel housing, the right column for a sensor mounted at the workpiece support.*

itself illustrates many discontinuities. The assignment of this DFC track to the grinding wheel rotational speed will be used in Subsection 4.3.2 for the sake of grinding wheel imbalance detection.

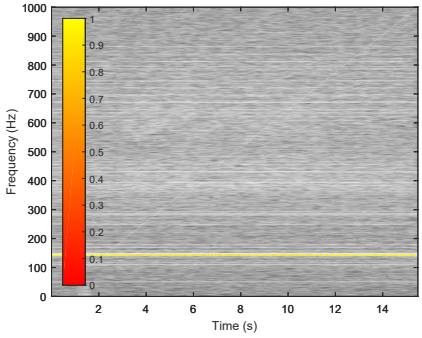
In a similar manner, one can upper bound the region of permissible control wheel rotational speeds by 500 rpm = 8.33 Hz. However, DFCs related to the control wheel are detectable neither for the acceleration sensors attached to grinding wheel housing nor workpiece support due to mechanical decoupling of the control wheel from both these machine parts. For an acceleration sensor directly attached to the control wheel housing, one might detect the DFC related to the control wheel rotational speed in a similar way as described above for the grinding wheel rotational speed.

Detection of Imbalances in Rotating Machinery

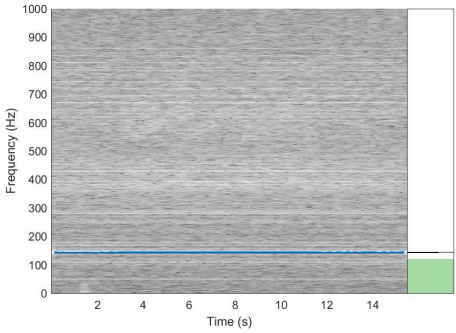
The former paragraphs illustrate, that assigning DFCs tracked during start-up to related rotating parts can be difficult, as DFCs often start to appear simultaneously. Only the grinding wheel DFC was identified from the multitude of DFCs leveraging upper bounds on permissible rotational speeds. Building on this assignment between grinding wheel rotational speed and its related DFC track, a workflow for constructing tailor-made features for imbalance detection is illustrated in this subsection. The same workflow is subsequently applied for detection of dressing wheel imbalance detection. These workflows thus cover the third major question formulated in the beginning of Section 4.3.

Imbalanced Grinding Wheel In order to understand the effect of imbalances in grinding wheels on recorded sensor data, spectrograms and related DFCs for both a balanced and an imbalanced grinding wheel are visualized in Fig. 4.8. The region of permissible grinding wheel rotational speeds is again marked by green patches in Figures 4.8b and 4.8d. While for the balanced grinding wheel (top row) only a dominant DFC at 143 Hz already observed during start-up is visible, the dominant frequency for an imbalanced grinding wheel is relocated at the 50 Hz component related to the grinding wheel rotational speed. Both dominant DFCs in Figures 4.8b and 4.8d are represented by a single track (illustrated in blue). This single continuous track is caused by amplitude estimates of constant height as visualized by the constantly yellow peaks in Figures 4.8a and 4.8c.

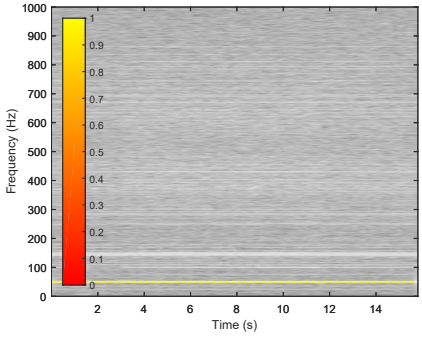
The dominant DFC at 50 Hz can be understood when considering the physical nature of (static) imbalances of the grinding wheel: For a balanced grinding wheel, the center of mass of the wheel and its geometrical center (i.e., its axis of suspension) are co-located. For an imbalanced grinding wheel however, the force applied from the grinding wheel to the grinding wheel housing is no longer temporally constant but becomes dependent from the relative position of the center of mass to the geometric center. More specifically, the applied force oscillates with the grinding wheel rotational speed. This results in a modulation of the envelope of the raw sensor signal, which in turn can be observed as a dominant DFC in the frequency domain.



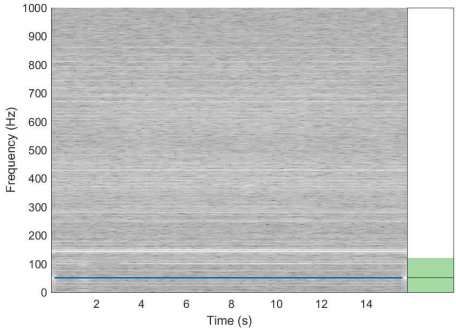
(a) Parameter estimates, sensor at grinding wheel housing, balanced wheel



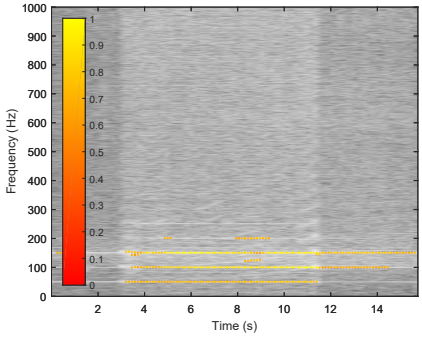
(b) Frequency component track, sensor at grinding wheel housing, balanced wheel



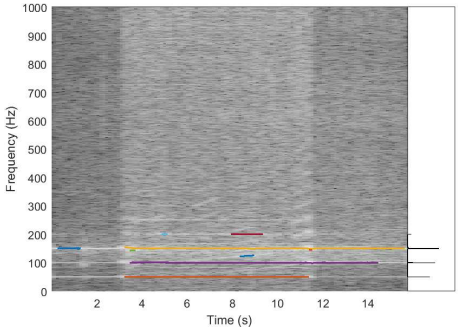
(c) Parameter estimates, sensor at grinding wheel housing, imbalanced wheel



(d) Frequency component track, sensor at grinding wheel housing, imbalanced wheel

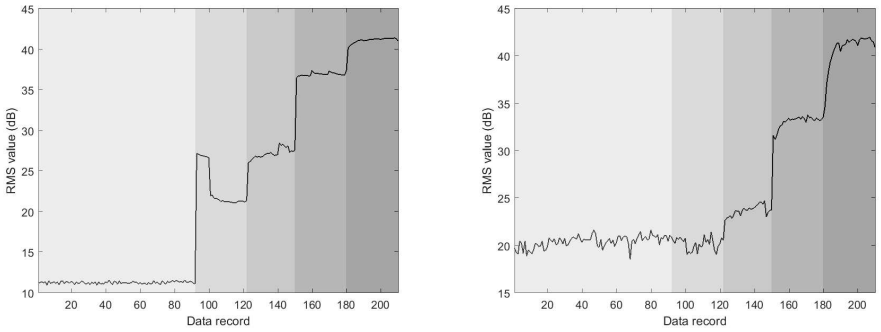


(e) Parameter estimates, sensor at workpiece support, imbalanced wheel



(f) Frequency component tracks, sensor at workpiece support, imbalanced wheel

Figure 4.8: *Parameter estimates (left column) and DFC tracks (right column) for a balanced (top row) and an imbalanced (middle and bottom row) grinding wheel. For the balanced wheel, a DFC at 143 Hz dominates. For the imbalanced wheel, TF energy is dominated by its rotational speed of 50 Hz (grinding wheel housing sensor, middle row) and related harmonics (sensor at workpiece support, bottom row).*



(a) RMS of grinding wheel rotational speed, sensor at grinding wheel housing

(b) RMS of grinding wheel rotational speed and first two harmonics, sensor at workpiece support

Figure 4.9: *Two exemplary features for detection of imbalances: RMS value for grinding wheel rotational speed (50 Hz, left) and RMS value for grinding wheel rotational speed and harmonics (right). Gray shadings depict different degrees of grinding wheel imbalances.*

This is illustrated in the middle row of Fig. 4.8: The dominant frequency relocates from the 143 Hz component to the rotational speed at 50 Hz. This suggests imbalance-related features depending on the energy of this rotational speed. Features building on increased energy at the grinding wheel rotational speed allow for detection of imbalances in the grinding wheel before machining workpieces, i.e., before the grinding wheel imbalance affects the quality of machined workpieces.

For spectrograms of acceleration sensors attached to the workpiece support (bottom row), DFC tracks at the grinding wheel rotational speed of 50 Hz and harmonics at 100 Hz and 150 Hz are observable. Similar to the explanation of the force applied from imbalanced grinding wheels to the grinding wheel housing described above, the force applied from the grinding wheel to the workpiece is no longer temporally constant. However, forces here do not behave purely sinusoidal. Thus, harmonics equidistant to the grinding wheel's rotational speed evolve. This suggests features related to these harmonics' energies.

In Fig. 4.9, two exemplary features related to the energy of the grinding wheel rotational speed (for the acceleration sensor attached to the grinding wheel housing, Fig. 4.9a) and related to the energies of this rotational speed and its harmonics (for the acceleration sensor attached to the workpiece support, Fig. 4.9b) are illustrated. The features are RMS values for TD signals where other DFCs than the grinding wheel rotational speed or its harmonics are filtered out. Both figures depict feature scores for data records with successively increasing degrees of imbalance severity (0, 0.5, 1, 3 and 6 μm). The different degrees of imbalance severity as illustrated by the differences in gray shading are clearly observable in the feature scores.

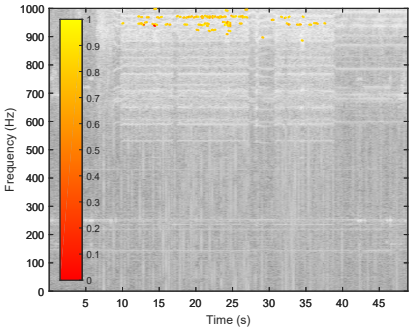
Imbalanced Dressing Wheel Similar to the approach sketched in the previous section, imbalances in other rotating machine parts can be detected. In Fig. 4.10, the effect of an imbalanced dressing wheel on the related TFDs of acceleration sensor signals is depicted. The acceleration sensor was attached to the dressing wheel motor (cf. Fig. 4.2). Similar to the effects of an imbalanced grinding wheel, a single dominant DFC at 58.33 Hz becomes apparent. This dominant DFC is related to the rotational speed of the dressing wheel. The DFC can be identified from the marginal histogram illustrated in Fig. 4.10d. Building on this dominant DFC at 58.33 Hz related to the dressing wheel rotational speed, custom-built features for a detection of dressing wheel imbalances can be defined: Similar to the features proposed for imbalanced grinding wheels in the previous subsection, RMS values are extracted from TD signals where other DFCs than the dressing wheel rotational speed are filtered out. Feature scores for balanced dressing wheels (light gray) and imbalanced dressing wheels (dark gray) are visualized in Fig. 4.11. The difference in feature scores verifies the relevance of this feature for dressing wheel imbalance detection.

Detection of Workpieces with Insufficient Roundness

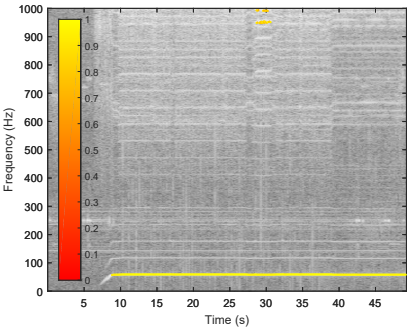
In addition to detecting imbalances in rotating machinery, detection of machined workpieces with an insufficient roundness is a predictive task where the detection of DFCs is relevant. As outlined in [97], workpieces with an insufficient roundness typically occur due to an instable choice of the grinding gap geometry, i.e., the arrangement of the workpiece in between grinding wheel, control wheel and workpiece support. For centerless grinding machines, the grinding gap geometry is mainly influenced by the height of the workpiece support h_w and its support angle β, i.e., the angle of elevation of the workpiece support. $\beta = 0$ means the workpiece support surface is horizontal. The grinding gap geometrical parameters h_w and β are visualized in Fig. 4.12.

When choosing an insensible combination of h_w and β, the machining process becomes unstable, resulting in an increasing oscillation of workpieces on the workpiece support. This oscillation results in higher amplitudes of undesired modulation of the workpiece surface. In machine tool literature, the modulation of the workpiece surface is typically referred to as *polygon surfaces* [125]. Such polygon surface modulations result in an appearance of harmonics of the workpiece rotational speed in the time frequency domain. The specific combination of h_w and β determines the actually observable order and amplitude of polygons and thus harmonics. In general, a superposition of different polygons / harmonics occurs.

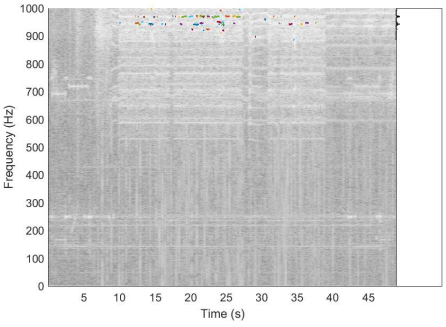
Typically, small degrees of modulation of the workpiece surface appear for all choices of process parameters h_w and β. The goal of finding an optimal choice of h_w and β is establishing a stable machining process, which leads to a maximum reduction of the existing roundness error during machining of workpieces such that predefined minimum requirements on roundness errors are reached. Reducing the roundness error has been shown to be mainly influenced by reducing the amplitudes at polygons of lower order [125].



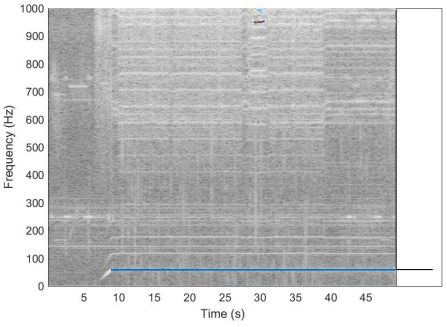
(a) Parameter estimates, dressing wheel housing, balanced wheel



(b) Parameter estimates, dressing wheel housing, imbalanced wheel



(c) Frequency component tracks, dressing wheel housing, balanced wheel



(d) Frequency component tracks, dressing wheel housing, imbalanced wheel

Figure 4.10: *Parameter estimates (top row) and detected DFC tracks (bottom row) for a balanced (left column) and an imbalanced dressing wheel (right column). For balanced dressing, the TF energy is dominated by high-frequency TFD regions. For the imbalanced dressing wheel, the TF energy is concentrated at a dominant DFC track of 58.33 Hz, which is related to the rotational speed of the dressing wheel.*

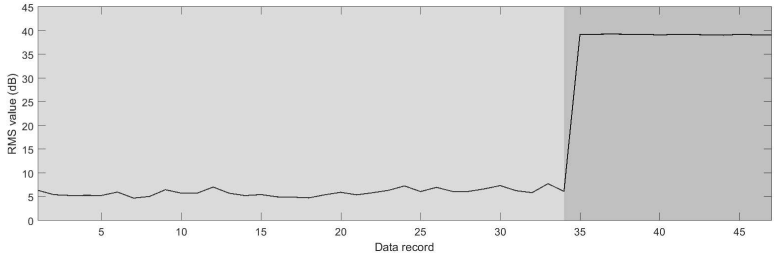


Figure 4.11: *Exemplary feature for detection of dressing wheel imbalances: RMS value for dressing wheel rotational speed. Gray shadings differentiate between a balanced dressing wheel (light gray) and an imbalanced dressing wheel (dark gray).*

In order to study the effects of instable process parameter choices on the roundness error of workpieces and energy distribution in TFDs, the height h_w of the workpiece support was deliberately raised by 1 mm compared to the stable normal height, which was the maximum possible height for the workpiece support without risking severe machine damage.

In order to verify the occurrence of harmonics of the workpiece rotational speed in the TFDs, the rotational speed of the workpiece f_{wp} has been computed using values of control wheel rotational speed $f_{ctrl} = 0.667\,Hz$, control wheel diameter $d_{ctrl} = 237\,mm$ and workpiece diameter $d_{wp} = 3\,mm$:

$$v_{ctrl} = f_{ctrl} \cdot \pi \cdot d_{ctrl} = 0.667\,s^{-1} \cdot \pi \cdot 237\,mm = 496.620\,mm/s \tag{4.25}$$

$$f_{wp} = \frac{v_{wp}}{d_{wp} \cdot \pi} = \frac{v_{ctrl}}{3.084\,mm \cdot \pi} = 51.258\,Hz \tag{4.26}$$

Here, the fact that circumferential velocities of control wheel v_{ctrl} and workpiece v_{wp} are the same was used. Building on the workpiece rotational speed and its harmonics, one can design tailor-made process monitoring features similar to the approach outlined in the previous subsections: Extracting the RMS value at harmonics related to the workpiece rotational speed allows to detect the change in energy at these frequencies due to machining workpieces with an insufficient roundness.

In Fig. 4.13, TFDs for the standard grinding gap geometry (left column) and the increased height h_w of the workpiece support (right column) are illustrated. Sensor data were recorded with the acceleration sensor attached to the workpiece support. As observable, the energy at already existent DFCs at 143 Hz and related harmonics at 286 Hz and 429 Hz increase. These components were already observed during start-up of the machine (cf. Fig. 4.7d) and are thus no harmonics of the workpiece rotational frequency. The reason for not being able to identify harmonics of the workpiece rotational speed could lie in the deliberate change of workpiece support height h_w not producing a sufficiently instable grinding gap geometry in order to effect lower-order polygon surfaces and thus harmonics

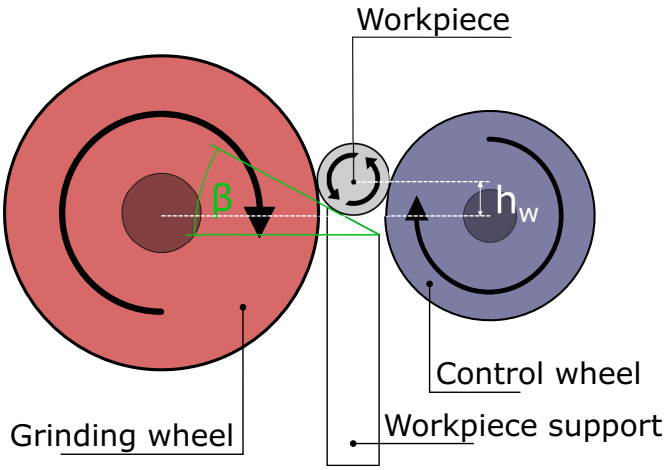
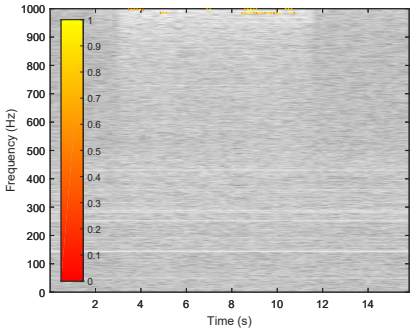
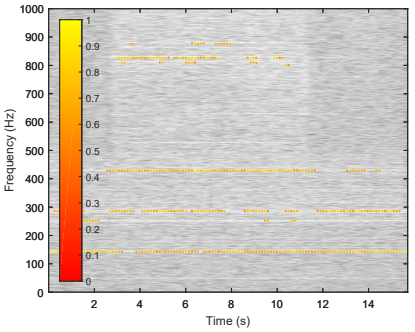


Figure 4.12: *Geometrical parameters h_w and β of the grinding gap*

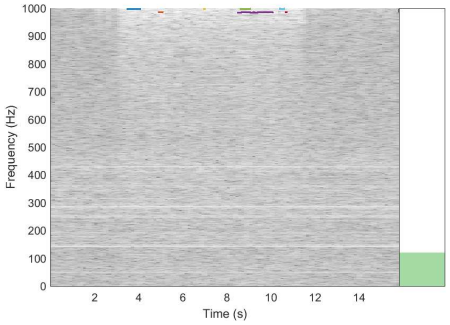
with high amplitudes. This assumption was confirmed by optical measurements, which illustrated no relevant roundness error of the measured workpieces. Thus, although the approach of tracking harmonics effected by workpieces with an insufficient roundness and subsequent extraction of features related to the energy at these harmonics might in general be valid, it can not be validated empirically here.



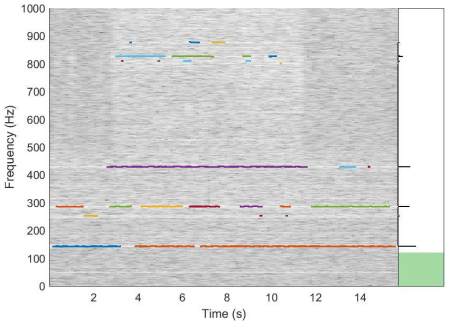
(a) *Parameter estimates, workpieces processed with normal h_w*



(b) *Parameter estimates, workpieces processed with increased h_w*



(c) *Frequency component tracks, workpieces processed with normal h_w*



(d) *Frequency component tracks, workpieces processed with increased h_w*

Figure 4.13: *Parameter estimates (top row) and detected DFC tracks (bottom row) for normal processed workpieces (left column) and workpieces processed with a deliberately increased height h_w of the workpiece support (right column). The spectrograms are computed for data measured with the acceleration sensor attached to the workpiece support. For normal processing, the TF energy is dominated by high-frequency TFD regions. For an increased workpiece support height, the TF energy at a DFC of 143 Hz and its harmonics increases.*

4.4 Conclusions

This chapter presented condition monitoring features for specific rotating machine parts. Features were designed based on DFCs and assigned rotating machine parts. Assigning a single DFC to its matching machine counterpart proved challenging, as a multitude of DFCs is observable during every operation state of the machine. Thus, additional meta information was used to constrain the search space on possible DFC candidates for assignment to a specific machine component. This meta information was given by upper bounds on maximum permissible rotational speeds (e.g., of a grinding wheel) and the non-stationary behavior of DFCs during start-up of the machine tool (which allowed to sort out persistently constant, spurious DFCs). Leveraging these additional constraints led to convincing results for the task of detecting imbalances in rotating machinery.

In general however, the approach proved both vulnerable to considering spurious DFCs during assignment and highly dependent on the capability to detect the DFCs of machine parts in the computed TFDs. The former was illustrated for detecting workpieces of insufficient roundness, where spurious DFCs were located in a similar region as harmonics related to the workpiece rotational speed are expected. The latter proved especially challenging for low-frequent DFCs and when the sensor from which measurement data was evaluated is not attached to the machine part of interest (e.g., control wheel rotational velocity not detectable with sensor attached to grinding wheel housing).

In summary, knowledge of DFC tracks assigned to rotating machine parts like grinding wheels and dressing wheels allows to design tailor-made features for imbalance detection. However, the condition monitoring approach presented in this chapter necessitates additional meta information during the assignment step. Even when such additional meta information can be formulated, the proposed approach is still vulnerable to spurious DFCs. In addition, handcrafted design of condition monitoring features in the proposed way necessitates a detailed understanding of machine tools and the parts they are assembled from.

Several adaptations of the approach visualized in Fig. 4.3 for increasing the quality of parameter estimation or DFC tracking would be possible: Regarding an automated way to choose the peak picking threshold during parameter estimation (e.g., based on the difference of distribution of TFD magnitudes for noise and signal samples) or a DFC tracking approach being able to deal better with noisy signals (e.g., based on computationally more expensive Markov renewal processes that allow modeling gaps in consistent tracks as “sleep states” [243], thus reducing the effect of fragmentation of component tracks discussed in Subsection 4.3.1). However, extracting features from DFCs still remains vulnerable to spurious components and still struggles with assigning a single DFC from a multitude of possible candidates to a dedicated rotating machine part.

In part II of this thesis, complementary approaches on including domain expertise into the detection of machine tool anomalies are presented. These approaches aim at collecting low-cost but high-quality annotations for the sensor data from domain experts and learning of anomaly detection models directly from the sensor data, i.e., without handcrafted design of tailor-made features.

Part II

Low-Cost Annotation and Robust Detection of Generic Machine Tool Anomalies

In part I of this thesis, a focus was put on designing tailor-made features for specific process monitoring tasks (e.g., tool condition monitoring) and condition monitoring tasks (e.g., imbalance detection in rotating machine parts) by explicitly modeling domain expertise in dedicated preprocessing algorithms (i.e., signal segmentation and estimation of discrete frequency components). This approach allows designing computationally simple but expressive features, custom-built for these tasks and generalizing to data recorded for other workpiece/parameter settings due to explicitly modeling physically understood cause-effect relationships. The approach, however, comes with several drawbacks:

- Concrete monitoring tasks have to be specified a priori in order to deliberately provoke the related anomaly types during dedicated measurement campaigns. This is often not possible.
- In addition, large data sets for different workpiece/parameter settings have to be collected and analyzed by data scientists in cooperation with domain experts. Afterwards, features have to be hand-engineered building on the cause-effect relations found and understood from data analysis. This involves a high human effort and thus high costs. In addition, the expertise of various groups (both domain experts and data analysts) is necessary in order to create such features, which might not be available (e.g., small production companies typically do not employ data analysts).
- Finally, provoking anomalies by deliberately choosing insensible machine parameter settings involves risking consequential machine damage.

Detection and labeling of anomalies during normal machine operation yields an alternative approach. Potential anomalies found by the anomaly detection models are proposed to domain experts for annotation. The resulting feedback (confirmation/rejection) yields labels for these proposed data records. Due to a concentration on potential anomalies, the labeling effort and thus cost is highly reduced: Machine tools on real production floors are typically well-regulated, resulting in only few data records representing abnormal machine behavior. Such human-in-the-loop collection of expert labels for reported anomalies constitutes a complementary approach of introducing domain expertise into the time series classification pipeline in Fig. 1.4 to the methods in part I of this thesis.

Despite generating labeled data sets for training of task-specific models, the obtained labels can be used to create semi-supervised extensions of the anomaly detection models. In machine monitoring scenarios, the benefit of including labels can be reasoned by the necessity to distinguish (often frequent) process parameter adjustments from (rare) real anomalies, as both parameter adjustments and anomalies result in signals deviating from the normal state.

In Chapter 5, a live and in situ annotation approach which allows to collect "in the wild" and low-cost labels for sensor data streams as outlined above is proposed. The approach is summarized in Fig. II.1. Building on a suitable representation of the raw sensor data (e.g., envelope signals, hand-engineered features or data representations learned via neural network models), data records under review are evaluated by a generic anomaly detection model regarding their degree

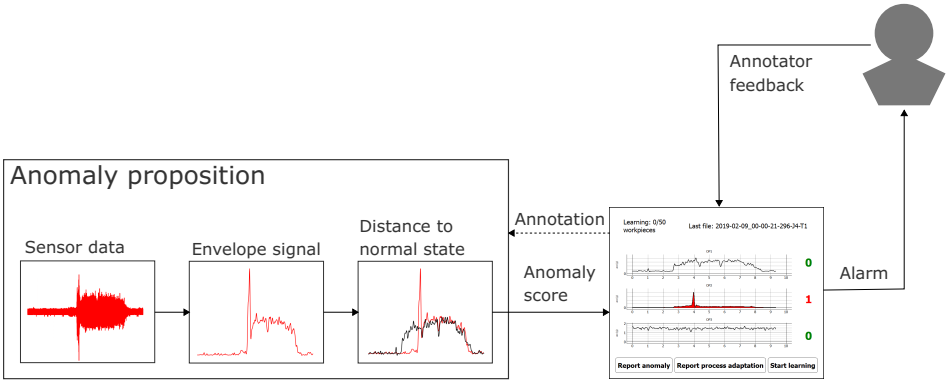


Figure II.1: *Visual summary of the live annotation approach*

of outlierness compared to formerly observed normal data or a representative model of the normal data. The degree of outlierness is measured by an anomaly score. For high anomaly scores exceeding a predefined threshold, a prototypical live annotation system which is attached to the outside of the machine tool can generate both a visual and acoustic alarm signal. This prototypical annotation system was developed in the context of this thesis. The visual and acoustic alarm triggers human feedback, i.e., the data record under review is annotated by the domain expert (machine operator).

In addition, Chapter 5 presents a proof-of-concept study regarding the validity of this approach. Building on the evaluation of obtained annotations and annotator behavior, advantages and disadvantages of the approach are discussed. The major findings suggest that live annotations are only in some cases of comparable or better quality than retrospective annotations by domain experts. Among possible explanations is the suspicion that the simple anomaly detector applied in the proof-of-concept study does not model normal behavior of the machine (as represented by the recorded data) sufficiently well, resulting in many false positives. Thus, in Chapter 6, advanced unsupervised neural anomaly detection models are discussed, which capture this normal behavior of the grinding machine better, even when the appearance of normal data evolves throughout the course of data recording. In addition, semi-supervised extensions of these neural anomaly detection models are studied. Both unsupervised and semi-supervised neural anomaly detection models are compared to the simple anomaly detector applied in Chapter 5 regarding their capability of more reliably proposing data records as abnormal.

Furthermore, a third alternative labeling approach (i.e., in addition to retrospective and live annotations by domain experts) is presented. This third approach allows to automatically generate labels from domain heuristics and hand-engineered features. Such automatically generated labels represent a second low-cost labeling alternative next to live annotations. Finally, semi-supervised neural anomaly detection models trained both with high-cost retrospective domain expert labels and low-cost automatically generated labels are compared.

5

User Study: Quality of Live Annotations and Influencing Factors

Disclaimer: The contents of this chapter have been published as [201].

This chapter presents a proof-of-concept study on the live and in situ annotation approach as summarized and visualized on the previous pages: Annotations for sensor data are collected directly (live) during their recording via a self-developed annotation prototype. This annotation prototype is attached to the outside of a demo grinding machine in a real-world production environment and allows considering both the current sensor data and additional meta information about the current machine state and workpiece quality which can be obtained by visual and auditory inspection of machine and workpieces (in situ).

First, details about the setup for data measurement (Section 5.2) as well as the design process and functionality of the proposed labeling prototype (Section 5.3) are described. Afterwards, methods for the evaluation of live annotations are discussed: Several assumptions for the evaluation of quality of the human label feedback collected via this labeling prototype are introduced in Section 5.4. These assumptions address the challenges of rating label feedback quality without being provided either reliable ground truth labels as gold standard for comparison or more than one live annotation per data record (in order to rate agreement among annotators). Then, results for the experiments conducted in order to select an appropriate anomaly proposing model and in order to rate the quality of labels collected via the proposed live annotation approach are stated in Section 5.5. The latter evaluation of live annotations is guided by the assumptions formulated before. Finally, the results are summarized and the strengths and weaknesses of the proposed live annotation approach in comparison to retrospective annotation are discussed critically.

5.1 Motivation

While a large amount of studies on collecting rare event labels in medical or social applications exists, studies concerned with annotation in industrial manufacturing surroundings are rare. Collecting labels for the rare anomalous events in such manufacturing surroundings is notoriously difficult. Often, frequent spurious signal outliers dominate seemingly detected anomalies and shadow the few real anomalies. This is even more difficult when anomalies are characterized by more subtle signal deviations than these spurious signal outliers. Depending on the chosen anomaly detection algorithm, this dominance of spurious outliers typically results in either a high false positive rate (FPR) or high false negative rate (FNR). This is even more the case for purely unsupervised models.

In the chosen machine tool monitoring application, spurious outliers are given by frequent process adaptations while real anomalies are typically rare. The reason for the latter is that machines in a real-world production surrounding are typically used for processing the same type of workpiece over a long period of time spanning several months to years. Thus, robust process parameter settings are known due to the well-understood machine behavior for this exact workpiece type, which in turn results in anomalies appearing only rarely.

In order to train anomaly detection models for a subset of specific known anomalies (e.g., imbalances in rotating machine parts, wear of ball screw drives or spindles), one can intentionally choose insensible process parameters to provoke these types of anomalies. Then, dedicated measurement campaigns for these anomaly types allow collecting labels and studying how these types of anomalies manifest regarding change of signal behavior. This approach was applied for the evaluations described in Part I of this thesis. The approach comes with short measurement campaigns (as the precious anomalous labels can be provoked intentionally) and thus only a small amount of additional costs due to loss of production time. Furthermore, high-quality ground truth labels can be obtained for these anomalies as the anomaly-causing machine parameters are under control. Several drawbacks arise:

- Provoking anomalies is still expensive, as retooling the machine for these provocations can be time consuming. Furthermore, precious production time is lost as the anomalously processed workpieces cannot be used after the experiment. Thus, annotating data sets with anomaly labels via dedicated measurement campaigns always comes with a trade-off: The higher the amount of labeled data, the better the performance of (semi-)supervised classifiers trained with such labels but also the higher the loss in production time and thus increase in costs.

- Many anomalies cannot be provoked intentionally, either due to unknown cause-effect relations of these anomalies or due to severe risks of consequential machine part damages.

- If anomalies can be provoked intentionally, the anomalies do not emerge in a natural way. As it is often non-trivial to distinguish between cause and

effect in the signal behavior, it is unclear whether the studied abnormal behavior generalizes to real-world anomalies.

- Finally, only anomaly types known in advance can be provoked.

Thus, collecting data and corresponding annotations "in the wild" (as opposed to an artificial provocation of anomalies) has the potential to yield more realistic labels. The disadvantages of labeling in the wild are high costs due to long data collection campaigns (as a high fraction of measured data does not illustrate anomalous machine behavior) and domain expert labeling afterwards. Furthermore, precious context knowledge for data annotation represented by the machine behavior during data collection is lost.

In this chapter, a third alternative approach is proposed: Prompting anomalous events to the machine operators for label feedback directly during everyday processing of workpieces. This live annotation approach allows collecting anomaly labels in the wild for low costs, as one does not have to rely on separate measurement campaigns but collects data during normal operation of the machine tools. Furthermore, the possibility to visually and auditory inspect the machine in situ gives the machine operators valuable additional information during live annotation of the collected data. Limitations to this approach are anticipated by the necessity of giving timely feedback to proposed anomalies (i.e., reduction of label quality due to time pressure).

For the proof-of-concept study presented in this chapter, a grinding machine in a real-world production surrounding was equipped with multiple MEMS vibration sensors for long-term measurements. In addition, both hardware and software of a labeling prototype, including the design of a suitable graphical user interface (GUI), for live and in situ annotation of sensor data records were developed and integrated. This physical prototype device was attached to the outside of the machine and connected to these sensors. This is visualized in Fig. 5.1.

Potential abnormal events are detected by a generic anomaly detection model. The anomaly detection model can then rise an alarm (both acoustically and by activation of a flash light) to trigger feedback of the human machine operator to the proposed anomaly. The visualization of sensor data records at the prototype comes with a GUI which guides the labeling process and additionally allows for user-initiated labeling of anomalies and process adaptations. Thus, the goal is to achieve a large data set of several weeks of sensor data records and related labels annotated by domain experts directly in the setting they were recorded.

The major challenge of this approach from an algorithmic point of view lies in the choice of an appropriate generic anomaly detection models. Guided by theoretically formulated constraints given by the embedded nature of the proposed system, the characteristics of the data and low latencies required by the application, tests on a labeled subset of the data are performed for an initial choice of anomaly detection models. The best-performing algorithm is then chosen for deployment on the proposed demonstrator system.

From a human-machine interface point of view, estimating the reliability both of anomaly propositions of the chosen anomaly detection model and of human label feedback is challenging due to the fact that for most of the data no ground

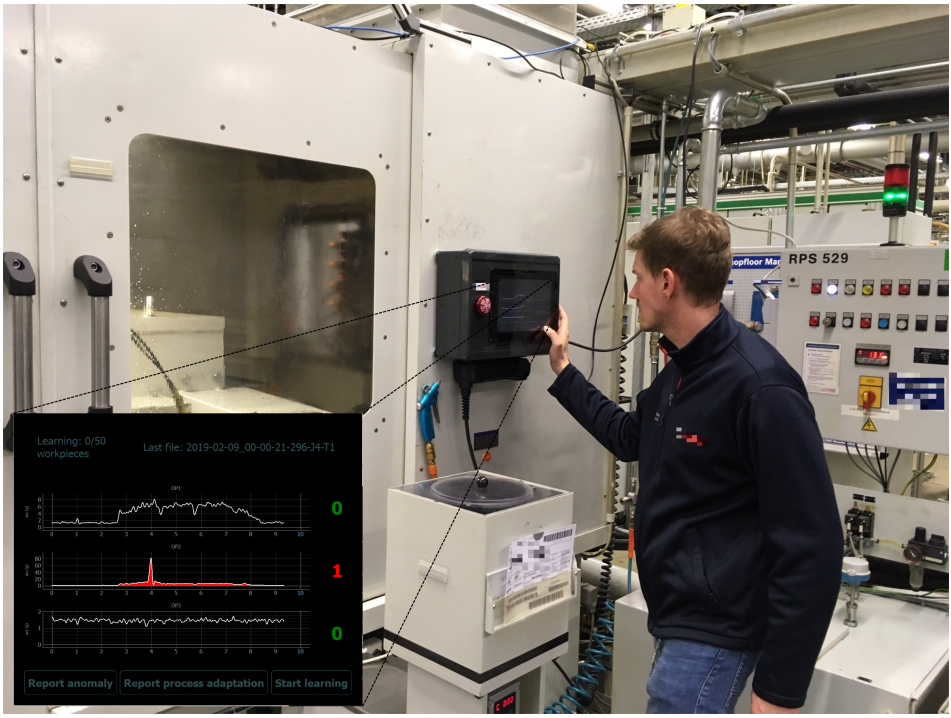
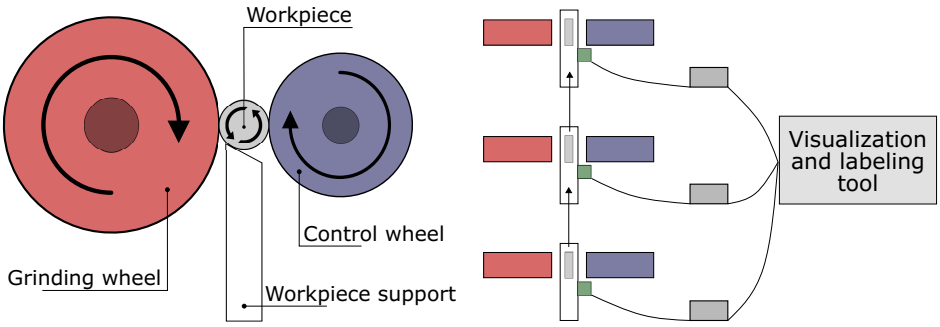


Figure 5.1: *Live and in situ annotation of sensor data records: A self-developed prototype visualizes sensor data records and allows for live and in situ annotation of anomalies (reactive to predictions by anomaly detection algorithms or initiated by the machine's operator). An exemplary screenshot of the default screen is illustrated in the lower left corner.*

truth labels exist. Furthermore, one cannot rely on comparison of labels from multiple annotators as typical crowd labeling methods do, because label feedback is collected from a single annotator (i.e., the current machine operator). Several assumptions both on label reliability and annotator motivation are introduced and validated. For this, the amount and distribution of label mismatch between anomaly propositions and live label feedback, labeling behavior of different annotators (inter-annotator agreement) during a second retrospective signal annotation phase and temporal evolution of labeling behavior of annotators are evaluated. Furthermore, the influence of certainty of the anomaly detection algorithm of its anomaly propositions (measured in height of anomaly scores), the familiarity of machine operators with the labeling user interface and other measures regarding user motivation on the reliability of live label feedback are investigated.

In summary, the main questions which are aimed to address in this chapter are as follows:

- Can high-quality but low-cost labels for machine tool anomalies be gener-



(a) *Process of external cylindrical grinding. Figure inspired from [241]*

(b) *Positions of mounted sensors*

Figure 5.2: *Left: Basic parts of a centerless external grinding machine. Right: Positions of mounted sensors at the grinding machine in this study. Three separate grinding/control wheel pairs allow for efficient machining of complex workpieces with successive processing steps.*

ated by incorporating machine operators' live label feedback to anomalies proposed by a generic unsupervised anomaly detection algorithm?

- Can a sensible and understandable human-machine interface for the live labeling prototype be developed by taking the opinion of end users (i.e., machine operators) into account during the design process?
- Can simple anomaly detection models respecting hardware constraints of the proposed embedded labeling prototype yield sensible anomaly propositions?
- How does the reliability of label feedback depend on the type of anomaly, the kind of signal visualization and the clarity of proposed anomalies (measured in height of anomaly scores)?
- How can reliability of the annotator label feedback be measured sensibly without access to ground truth labels for most of the data and with label feedback for only one annotator at a time (i.e., the current operator of the machine tool)?

5.2 Measurement Setup

In this section, information about the measurement setup is presented. This includes specifications about the used sensor types and positions. All data were collected from the centerless external cylindrical grinding machine illustrated in Fig. 5.1 which was equipped with the proposed labeling prototype.

The data of this study were recorded using MEMS vibration sensors as described in Table 1.1. For the measurement of process-related anomalies, the

workpiece support as depicted in Fig. 5.2a proved to be a suitable sensor mounting position. The grinding machine examined in this study was rather complex and encompassed three workpiece supports. These allowed for three subsequent processing steps and thus machining of geometrically complex workpieces.

An overview of the measurement setup is illustrated in Fig. 5.2b. The three workpiece supports are depicted in white with workpieces depicted in gray. Grinding wheels and control wheels associated to the three successive processing steps are shown in red and blue, respectively. The successive processing of the workpieces starts on the bottom workpiece support, proceeds to the middle workpiece support and is finished on the top workpiece support. This processing order of the workpieces is indicated by the direction of depicted arrows. Each workpiece support is equipped with a sensor (green). The bottom sensor is termed OP1, the middle sensor OP2 and the top sensor OP3. The most relevant sensor positions for anomaly detection are OP1 and OP2, where most of the material removal from the workpiece happens. Each sensor is connected to an embedded PC (gray) acting as gateway system for local preprocessing and data handling. The gateway systems are in turn connected to the labeling prototype.

5.3 Description of the Visualization and Labeling Prototype

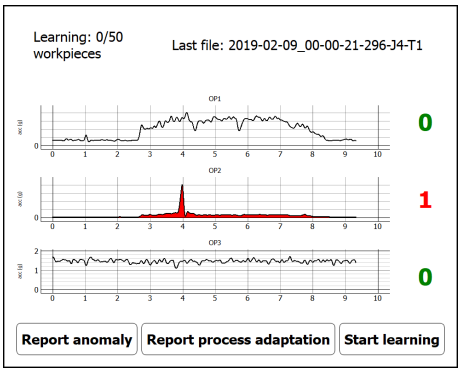
In order to understand the design considerations of the proposed labeling prototype, the characteristics of the labeling surrounding and how these are addressed during the design of the visualization and labeling prototype are described in this section. Furthermore, the intended use of the labeling prototype is sketched. The final design of the prototype is visualized in Fig. 5.3.

5.3.1 Design Process of the Labeling Prototype

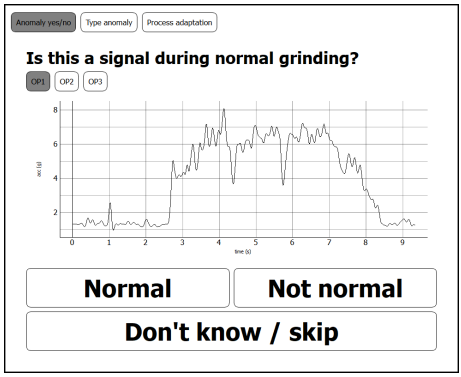
The design of the labeling tool evolved both through many interviews with the machine operators and design considerations deducted from the typical working conditions on the factory floor. The characteristics of the industrial surrounding and the design considerations with which these characteristics are aimed to be addressed can be summarized as follows:

- First general impressions of the surrounding included its loudness and the necessity of the machine operator to be capable of handling multiple tasks in parallel.

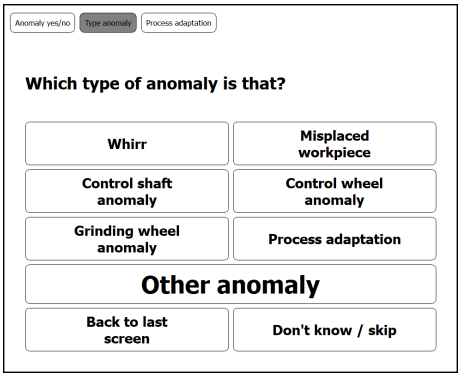
- In order to draw the attention of the machine operator to the labeling prototype display while being involved with other tasks, an alarm flash light and red coloring of proposed abnormal signals was triggered. Furthermore, an acoustic alarm signal was activated. This alarm signal had to be rather loud due to the noisy surrounding of the machine.



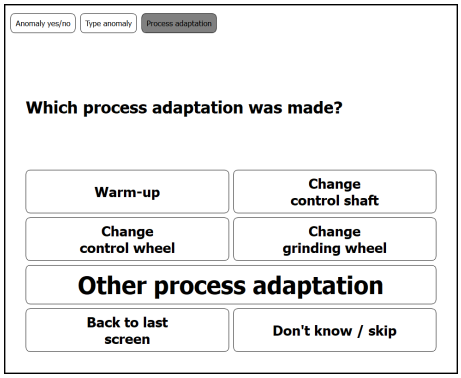
(a) Screen 1: Cont. visualization



(b) Screen 2: Anomaly (binary)



(c) Screen 3: Anomaly (multi-class)



(d) Screen 4: Process adaptation (multi-class)

Figure 5.3: *Screens of the visualization and labeling prototype (English version). The figures illustrate screenshots of the developed labeling prototype which was deployed on the factory floor.*

- To address the expected uncertainty of annotators in the annotation process which occurred due to handling multiple tasks in parallel, an opportunity to skip the labeling when uncertain was included (buttons "Don't know / skip" on screens in Fig. 5.3). In addition, switching between the successive labeling screens manually to review the visualized signals again during the labeling process was allowed (buttons "Back to last screen"). Finally, void class buttons ("Other anomaly" and "Other process adaptation") allowed to express uncertainty about the class of anomaly / process adaptation or giving a label for an anomaly / process adaptation which was not listed among the label choices.

In addition, the end users of the proposed labeling prototype (i.e., the machine operators and their shift leader) were included at multiple stages of the

design process in order to allow for a design of the labeling prototype guided by optimal user experience.

- In order to define an initial version of the labeling prototype screen design, a first meeting with the shift leader was arranged. In this meeting, a first version of the labeling prototype design was proposed and adapted. In addition, the most accustomed way for presentation of sensor data was discussed: Industrially established solutions typically depict the envelope signals rather than the raw sensor data, TFD representations or feature scores. Thus, this well-known form of envelope signal representation was chosen. Finally, the most frequent anomaly types and process adaptations to be included as dedicated class label buttons (screens 3 and 4 in Fig. 5.3) were discussed.

- After implementation of the labeling GUI from the adapted design of the initial meeting, the user experience of the proposed labeling GUI was discussed in a second meeting with the shift leader. This involved a live demo of the suggested labeling GUI in order to illustrate the intended use of the labeling prototype and resulted in a second rework of the labeling prototype.

- After this second rework of the labeling prototype, a meeting was arranged including both the shift leader and all machine operators. This meeting included a live demo of the labeling prototype directly at the grinding machine and a discussion of the terms chosen for the labeling buttons on screen 3 and 4 depicted in Fig. 5.3. In addition, an open interview gave the opportunity to discuss other ideas or concerns regarding the design or use of the labeling prototype.

- In order to address remaining uncertainties about the intended use of the labeling prototype after deployment at the demo grinding machine, a short instruction manual was written and attached next to the labeling prototype at the machine.

The final visualization and labeling prototype GUI is shown in Fig. 5.3. Background colors of the screens were changed to white (black on the original screens) for better perceptibility of visual details. The terms stated on the screens were translated verbatim to English in these figures for convenience of the reader. Apart from the translated terms and the change in colors, the screens depicted in Fig. 5.3 are identical to the original screens. The GUI with original background colors and language descriptions can be found in the appendix.

To the best of the author's knowledge, no previous work has focused on collecting data annotations via direct human feedback in industrial applications like described here. Furthermore, the industrially robust prototypical annotation tool used here is different from typical off-the-shelf smartphone or tablet devices and involves different design implications, which are described here for the first time.

5.3.2 Functionality of the Labeling Prototype

In this subsection, a brief overview of the intended use of the labeling prototype is presented. The default screen (screen 1) as depicted in Fig. 5.3a illustrates the sensor envelope signals.

When the anomaly detection algorithm detects an anomalous signal behavior an alarm is generated: The signal is colored in red, furthermore both an acoustic alarm and a flash light are activated and the anomaly counter to the right of the alarm-causing signal is incremented. By pressing this counter button, the user is guided to the second screen as shown in Fig. 5.3b. On this second screen, the user can review the alarm-causing signal and the signals of the other sensors by switching between the tab buttons "OP1", "OP2" and "OP3". If the signal is considered normal, the user can return to screen 1 by pressing the button "Normal". If the signal is considered abnormal, the user should press the button "Not normal" and is be guided to screen 3 as shown in Fig. 5.3c for specifying the type of anomaly.

On screen 3 then, the user is prompted a choice of the most typical anomaly types. A button "Other anomaly" allows specifying that either the anomaly type is not listed or that only vague knowledge exists that the signal is anomalous but that the type of anomaly is unknown. This button might for example be pressed in case of a common form of envelope signal that is known by the operator to typically appear before certain machine anomalies or by clear signal deviations with an unfamiliar signal pattern. By pressing the button "Back to last screen" the user can return to screen 2 for re-considering the potentially abnormal signal under review. By pressing the button "Process adaptation", the user is guided to screen 4 as illustrated in Fig. 5.3d, where the signal under review can be labeled as showing a process adaptation. The reason for this is that a generic unsupervised anomaly detection model can typically not distinguish between signal outliers due to a real anomaly or major process adaptations and might report both as a potential anomaly. On screen 4, the user is again prompted with a selection of most typical process adaptations and the possibility to specify "Other process adaptation" if the type of process adaptation is not listed.

On each screen, the user has the possibility to abort the labeling process by pressing the "Don't know / skip" button. This allows returning to the default screen (screen 1) when uncertain about the current annotation. Higher quality labels are assumed to be possible by these buttons allowing to express annotator uncertainty.

On screen 1, the user is given three more buttons for self-initiated activities. "Report anomaly" allows the user to specify an abnormal signal not reported by the anomaly detection models. These false negatives are the most precious anomalies as they are the ones that could not be detected by the anomaly detection algorithms. The button "Report process adaptation" allows reporting process adaptations, which both gives useful meta information for later signal review by the data analyst and allows to learn distinguishing between signal outliers due to (normal) process adaptations and anomalies. The button "Start learning" finally allows initiating a re-learning of the anomaly detection model. This button

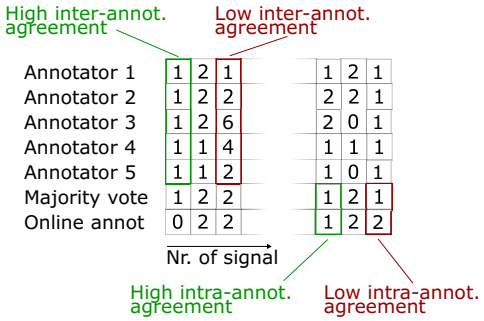


Figure 5.4: *Explanation of measures for inter-annotator agreement and intra-annotator agreement*

must be considered after major process adaptations or when the learning process was initiated during abnormal signal behavior, as then the learned normal machine behavior is not represented well and consequently results in frequent false positives. The state of learning is depicted by a counter in the upper left corner of screen 1, which allows the user to consider re-learning (i.e., if abnormal events occurred during learning) and in general makes the state of learning apparent to the user.

5.4 Assumptions on Evaluation Measures

In this section, the assumptions on evaluation of online label feedback which are introduced with this work are discussed.

5.4.1 Assumptions on Measures for Quality of Label Feedback

As mentioned in the former section, this study is confronted with the challenge of rating label reliability without access to ground truth labels. In addition, only one label feedback per proposed data record is received (assigned by the current machine operator), which makes rating reliability of online label feedback via inter-annotator agreement (consistency between labels of multiple annotators) impossible. Thus, alternative strategies and assumptions for rating reliability of online label feedback are imposed:

- Assumption 1: Reliable online annotations are assumed to coincide with a low mismatch between anomaly propositions of the anomaly detection model and online annotator feedback (i.e., a high confirmation rate). The amount of confirmed anomalies per class yields information about which types of anomalies can be identified reliably by annotators: Frequently confirmed anomaly types are assumed to be identifiable well from the sensor signals visualized with the proposed labeling prototype.

- The confirmation rate of a reliable online label feedback is assumed to be

dependent on anomaly scores and time of proposing data records for annotation.

  - Assumption 2a: Reliable label feedback is assumed to coincide with a high confirmation rate of data records with high anomaly scores (assigned by the anomaly proposing anomaly detection model), as anomaly scores reflect the degree of outlierness of signals under review. Thus, signals with high anomaly scores deviate more clearly from normal signals.
  - Assumption 2b: In addition, a higher degree of confirmed anomaly propositions is assumed to be observable for days with visually confirmed anomalies (e.g., due to machine inspection by the operators). On the other hand, if anomaly propositions for clearly outlying signals are rejected although anomalous machine behavior was confirmed afterwards, small reliability of this label feedback is assumed.

- For a high mismatch between anomaly proposition and online label feedback it is hard to decide whether proposition or feedback is more trustworthy. In order to still be able to assess reliability of online label feedback, a second period of retrospective signal annotation is introduced: Signals proposed as anomalous to the machine operators during online annotation are considered for a second review. Multiple annotators are then asked to inspect these signals again retrospectively. Comparison of online label feedback with this second set of retrospective labels allows rating inter-annotator agreement (i.e., consistency between retrospective labels of multiple annotators) and intra-annotator agreement (i.e., consistency of annotations between first (online) and second (retrospective) labeling period):

  - Assumption 3a: Reliable retrospective labels are assumed to coincide with a high inter-annotator agreement.
  - Assumption 3b: Reliable online label feedback is then assumed to coincide with a high intra-annotator agreement between online label feedback and (the majority vote of) retrospective labels (assumption 3b). The majority vote of the multiple retrospective labels per proposed signal has to be computed in order to make the single online label feedback comparable with multiple retrospective labels. A subject-specific annotator agreement cannot be computed, as access to shift plans cannot be granted (due to local data protection laws).

For a better understanding, different scenarios of inter- and intra-annotator agreement are visualized in Fig. 5.4. Here, retrospective annotators 1 to 5 are shown the signals proposed as anomalous during online annotation for a second review. Inter-annotator agreement can be judged from these 5 annotations per proposed signal. The majority vote found from these 5 annotations per signal is depicted in row 6 and allows for comparison of retrospective annotations with online annotations (row 7). This in turn allows for judging intra-annotator agreement, i.e., consistency between both labeling periods for each signal proposed as anomalous.

Additional to the assumptions stated above, high label reliability is related to high annotator motivation. Annotator motivation, on the other hand, is estimated by the assumptions stated in the following.

5.4.2 Assumptions on Measures for Annotator Motivation

- Assumption 4a: High annotator motivation is assumed for a high reaction rate during online annotation to labels proposed by the anomaly detection algorithm. The reaction rate is measured by the ratio of anomaly propositions which the annotator reacted to by either confirming an anomaly or rejecting the proposed label (by assigning a "Normal" label). Furthermore, an intentional skipping of the current anomaly proposition by pressing the "Don't know / Skip" button is rated as a reaction.

- Assumption 4b: A small reaction latency during online annotation to labels proposed by the anomaly detection algorithm is assumed to be a sign of high annotator motivation.

- Assumption 5: Finally, a high degree of user-initiated actions for days with visually confirmed anomalies is assumed to correlate with a high user motivation. This is due to a higher necessity of process adaptations after confirmed anomalies and a higher necessity of reporting anomalies missed by the anomaly detection model during time periods of abnormal machine behavior. The degree of user-initiated actions is measured by the number of clicks of any of the buttons for user-initiated actions on the proposed visualization and labeling prototype (cf. Fig. 5.3a, buttons "Report anomaly", "Report process adaptation" and "Start learning").

5.5 Experiments

This section presents experiments conducted both for the initial choice of an anomaly detection model and for evaluations regarding the assumptions imposed on label quality and annotator motivation in Section 5.4.

5.5.1 Selection of a Generic Anomaly Detection Algorithm

A sensible choice of anomaly proposing algorithm (i.e., anomaly detection model) had to be found among the rich potential choice of models. The anomaly detection model of choice should both fulfill requirements regarding predictive quality and address the computational constraints (restricted memory space, restricted computational time during predictions) arising from the embedded nature of the custom-built, deployed labeling prototype and the nature of the application (fast reporting of potentially high-risk anomalies).

Evaluation Data

For selection of a suitable anomaly detection algorithm, the challenge is how to measure predictive quality of models without reliable ground truth labels for the data. In fact, the very motivation of installing the visualization and labeling prototype at the grinding machine observed in this study was that critical process problems occurred at this machine but the cause of them remained widely unknown.

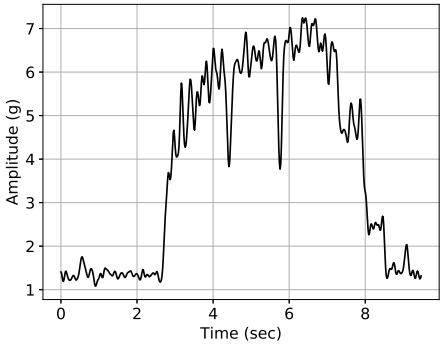
The data sets chosen for estimation of predictive quality of anomaly detection candidates (data sets DS1 and DS2) were recorded at two successive days with visually confirmed machine damages. Multiple successive workpieces were processed with a non-optimal interaction between grinding wheel and control wheel, which resulted in "whirring workpieces" and finally a damage of the grinding wheel. Whirring of workpieces is typically caused by the workpiece not being decelerated properly by the control wheel. A whirring workpiece is then accelerated to the speed of the grinding wheel and ejected from the workpiece support, flying through the machine housing - thus the term whirring workpiece. This type of anomaly is referred to as whirr anomaly in the following text.

DS1 and DS2 data were recorded during initial test measurements prior to the online annotation experiments involved with this study. The visual confirmation of machine damages allowed for a labeling of whirr anomalies and grinding wheel damages in discussion with the domain experts and can thus be interpreted as ground truth labels. DS1 (3301 data records, 293 anomalies) includes a higher proportion of anomalies than DS2 (3692 data records, 22 anomalies). Thus, predictive results for DS1 were assumed to be more informative regarding choice of an appropriate anomaly detection algorithm.

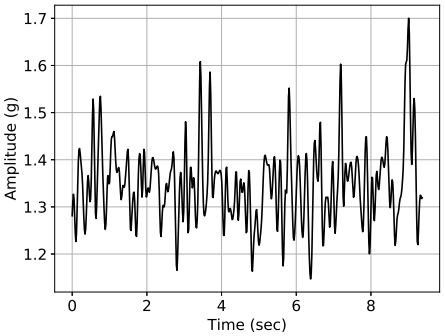
Exemplary signal envelopes for the different classes present in data sets DS1 and DS2 are illustrated in Fig. 5.5. An exemplary normal signal of sensor OP1 is depicted in Fig. 5.5a. The most severe class of anomaly at the considered grinding machine was whirring of workpieces. An exemplary signal is depicted in Fig. 5.5c. As mentioned above, whirring workpieces can result in severe damage of machine parts, especially of grinding wheel and control wheel. An exemplary signal of a visually confirmed damage in the grinding wheel due to multiple successive whirring workpieces is illustrated in Fig. 5.5d. Warm-up signals as depicted in Fig. 5.5b can be observed typically after machine part changes due to detected anomalies or when the machine is started after a longer down-time. Warm-up is the most frequent type of signal related to process adaptations. In order to create a binary classification scenario for selection of a generic (binary) anomaly detection algorithm, labels for all anomaly classes were merged into a single anomalous label class.

Anomaly Detection Models and Features

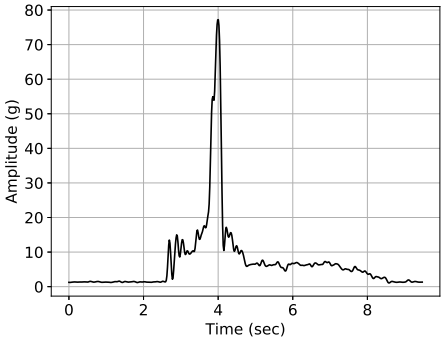
Additional to comparison of predictive quality of anomaly detection model candidates on labeled data sets DS1 and DS2 the following requirements for the choice of an anomaly detection algorithm can be formulated due to the constraints im-



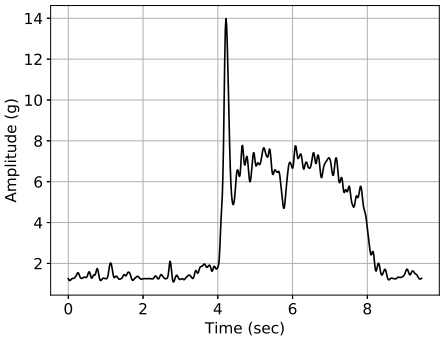
(a) *Normal (OP1), DS2*



(b) *Warm-up (OP1), DS2*



(c) *Whirr (OP1), DS1*



(d) *Damaged gr. wheel (OP1), DS1*

Figure 5.5: *Exemplary envelope signals for different classes of normal behavior (Fig. 5.5a), process adaptations (Fig. 5.5b) and anomalies (Fig. 5.5c, Fig. 5.5d). Damage of the grinding wheel occurred due to multiple, successive whirring workpieces.*

posed by data characteristics and embedded nature of the deployed labeling prototype:

- The algorithm is not provided with any labels during the conducted live annotation experiments and must thus allow for completely **unsupervised learning**.
- The algorithms must allow for **fast predictions** (due to computational constraints and the goal of creating timely alerts for potential anomalies) and have **low memory occupation** (embedded system with restricted memory space).
- Frequent process adaptations necessitate **fast re-learning** or **fast transfer learning** capabilities of the models in order to retain an appropriate representation of the normal state.

In Tables 5.1 and 5.2, results for comparison of different anomaly detection models on data sets DS1 and DS2 both regarding predictive quality (precision, recall and F1 score) and predictive cost (training time, prediction time, memory occupation) are stated. The predictive measures are stated as class-weighted scores, i.e., class imbalance is taken into account. Memory occupation is stated in kilo bytes, training time in seconds, prediction time in milliseconds. All experiments were evaluated on an *Intel Core i7-6700* with 3.4 GHz without any optimization of code or parallelization. The upper part of the tables are occupied by methods relying on one-dimensional data representations, the lower parts by methods relying on multi-dimensional (i.e., feature space) representations. For feature space methods, the implementations of scikit-learn [184] and PyOD [290] were used where available.

Most anomaly detection algorithms stated here rely on an assumption of the outlier fraction. The real outlier fraction was provided which was computed from DS1 and DS2 ground truth labels. For Half Space Trees (HSTrees), 100 estimators with a maximum depth of 10 were used. For xStream, 50 half-space chains with a depth of 15 and 100 hash-functions were used. All other parameters were chosen as the default values provided with the scikit-learn and PyOD implementations. For SDTW, $\gamma = 1.0$ was chosen as proposed in [63] due to their observation that DTW (which can be recovered by setting $\gamma = 0$) or SDTW with low γ values can get stuck in non-optimal local minima.

NC methods come with the necessity to specify a decision threshold between normal and abnormal behavior. This value was specified based on the Euclidean distances of envelope signals observed during training: First, a normal centroid was computed from training examples by Euclidean averaging of training envelope signals [63]. The anomaly detection threshold was then chosen as the mean plus $n_{std} = 10$ times the standard deviations of Euclidean distances of these training examples to the normal centroid. During prediction, Euclidean distances to the trained normal centroid were computed and compared to this threshold value in order to predict whether the current test envelope signal is normal or abnormal. As of now, these Euclidean distances of test envelope signals to the trained normal centroid are referred to as "anomaly scores". The normal centroid is kept up to date to the latest normal data by weighted averaging with the incoming envelope signals classified as normal. Optionally, envelope signals can be aligned via cross correlation before computation of the ED measure. Signal alignment yields translation invariance (TI) of envelope signals.

As mentioned in the related work chapter, multi-dimensional anomaly detection methods introduce the additional challenge to find a generic, expressive set of features. A set of features consisting of a combination of statistical features and wavelet-based features was chosen as these are both generic and prominent in many machine health monitoring applications [252]. Statistical time domain features consist of the first four central moments (mean, standard deviation, skewness and kurtosis). Wavelet-based frequency domain features were computed by a simple discrete wavelet transform for a db4 wavelet family base and a decomposition level of 8. This resulted in a 13-dimensional feature vector per each data record.

Results

The results in Tables 5.1 and 5.2 illustrate - in accordance with literature on time series classification (TSC) - that supervised 1NN and anomaly detection methods based on one-dimensional signal representations in general (i.e., the various compared NC models) are highly expressive. Furthermore, the high-quality predictive results confirm that envelope signals expose enough information for detection of the anomaly types present in data sets DS1 and DS2. The latter is in accordance with the observation of the capability of experienced machine operators to estimate non-optimal machine behavior for many anomaly classes from the typical envelope signals displayed for commercially available industrial sensors.

1NN performed only acceptable in a supervised setting, at least without further (clustering/density) assumptions being introduced. On the other hand, NC methods illustrated excellent predictive performance for both data sets DS1 and DS2. In this study, the ED measure was competitive to DTW and SDTW while resulting in faster training/prediction as stated in Tables 5.1 and 5.2.

NC models combined with ED measures (NC (ED)) performed especially well when signals were aligned to the normal centroid via cross correlation before computation of the ED measure (NC (ED+TI)). The reason for this is the nature of the presented data: Applying the same processing steps to each workpiece results in a highly similar envelope signal for each (normally) processed workpiece and thus in no need to warp signals before computation of distance measures as done via DTW. Signal alignment via cross correlation however yields a computationally efficient translation invariance of signals, which takes typical process adaptations (like changing the point in time of initial contact between grinding wheel and workpiece) into account. This in turn results in these signals during process adaptations not being falsely proposed as anomalies, thus effectively reducing the false positive rate.

While anomaly detection methods based on envelope signals performed well on both data sets, basic feature space methods failed to capture normal behavior especially for DS1. The reason for this is assumed to be given by the more complex anomalies present in DS1 than in DS2. Among feature space methods, only more advanced methods like HDBSCAN and streaming feature ensemble methods (LODA, HSTrees, RSForest, RSHash and xStream) illustrated a reasonable predictive quality. Nonetheless, these methods yielded worse predictive quality while occupying more memory and/or revealing longer prediction times than NC methods.

Table 5.1: *Comparison of anomaly detection models on data set DS1 (binary labels)*

Algorithm	F1 Score	Precision	Recall	Memory [kB]	Training Time [s]	Prediction Time [ms]
1NN (ED) sup.	99.85	99.85	99.85	8742	0.09	1.29
NC (ED)	99.75	99.75	99.75	23	0.05	0.05
NC (ED+TI)	99.80	99.80	99.80	23	0.78	0.47
NC (DTW)	99.20	99.22	99.19	21	1641.55	861.28
NC (SDTW)	99.30	99.30	99.30	21	205.99	1016.63
LOF [46]	52.32	88.45	47.25	170	0.05	0.03
CBLOF [96]	53.55	88.50	48.36	12	0.94	0.01
IF [146]	55.84	88.61	50.48	78	0.14	0.03
kNN	53.11	88.48	47.96	150	0.06	0.09
MCD [205]	67.45	89.37	62.24	19	0.61	0.01
OCSVM [220]	51.59	88.42	46.60	109	0.06	0.02
HDBSCAN [49]	96.26	96.61	96.47	599	0.20	0.01
LODA [188]	90.06	93.62	88.94	29	0.02	0.01
HSTrees [247]	96.43	96.75	96.62	278	6.31	5.12
RSForest [272]	96.15	96.31	96.31	304	4.82	5.10
RSHash [215]	95.92	96.12	96.11	1807	2.17	0.01
xStream [157]	96.25	96.45	96.42	246,994	13.00	8.04

Table 5.2: *Comparison of anomaly detection models on data set DS2 (binary labels)*

Algorithm	F1 Score	Precision	Recall	Memory [kB]	Training Time [s]	Prediction Time [ms]
1NN (ED) sup.	100.0	100.0	100.0	9675	0.06	1.44
NC (ED)	100.0	100.0	100.0	23	0.05	0.06
NC (ED+TI)	100.0	100.0	100.0	23	1.11	0.62
NC (DTW)	100.0	100.0	100.0	23	1676.78	785.73
NC (SDTW)	100.0	100.0	100.0	23	174.61	911.29
LOF [46]	99.41	99.60	99.32	190	0.05	0.04
CBLOF [96]	100.0	100.0	100.0	12	0.03	0.01
IF [146]	99.79	99.82	99.77	79	0.17	0.04
kNN	99.48	99.63	99.41	167	0.05	0.11
MCD [205]	99.75	99.79	99.73	20	0.66	0.01
OCSVM [220]	99.30	99.55	99.19	101	0.06	0.02
HDBSCAN [49]	99.98	100.0	99.01	669	0.35	0.01
LODA [188]	99.83	99.85	99.82	29	0.03	0.01
HSTrees [247]	99.65	99.68	99.68	278	6.75	5.08
RSForest [272]	99.71	99.73	99.73	304	4.95	5.14
RSHash [215]	100.0	100.0	100.0	2019	2.52	0.01
xStream [157]	99.81	99.82	99.82	224,335	15.17	9.35

Choice of Anomaly Detection Model for Deployment

In accordance with the requirements for an anomaly detection algorithm formulated at the beginning of Subsection 5.5.1, the NC model combined with the ED distance measure and signal alignment (NC (ED+TI)) was chosen to be deployed in the labeling prototype due to its excellent performance on data sets DS1 and DS2, the small and constant memory requirements as well as fast (re-)training and prediction times. Furthermore, this model states an intuitive anomaly score per each considered data record as described above, which is made use of in the following section on label evaluation results.

In order to allow for quick reaction in case of whirring workpieces, a simple threshold heuristic which yields an alarm signal when a prespecified signal amplitude threshold is exceeded was additionally deployed. This allowed generating timely warnings not only on the level of complete signals (as via the decision threshold of the NC model) but for each signal envelope sample. Furthermore, this amplitude threshold heuristic allowed for alarms during re-learning of the NC model.

Thus, the threshold heuristic was implemented mainly to allow for timely alarms of safety-critical whirring workpieces, even when the NC model was not available (i.e, during (re-)learning). However, parallel anomaly detection by both models additionally allowed comparing the simple threshold heuristic with the more advanced NC model (having the potential to judge anomalous behavior of signals both on sample and sequence level by taking signal forms into account). For whirr anomalies with their characteristic and well-understood high-amplitude peak pattern as illustrated in Fig. 5.5c, a good detection rate with the threshold heuristic was assumed. For subtle anomalies however, a better detection rate with the NC model was assumed. Furthermore, a smaller FP rate was assumed for the simple threshold heuristic as it only generates alarms for characteristic high-energy peak patterns (i.e., whirring workpieces), while the NC model also generates alarms for other more subtle anomalies (e.g., manifesting in small amplitude deviations in multiple signal locations or across complete signals). These subtle anomalies were assumed to be visually harder to identify by the machine operators, thus yielding a higher FP rate for the NC model. In general, among the most interesting questions regarding evaluation of online signal annotations collected via direct human feedback were:

- Can online annotations yield reliable signal labels (in comparison to retrospective annotations)?
- Which types of anomalies can a human annotator detect by reviewing sensor signal envelopes (both during online annotation and retrospective annotation)?
- Can human operators identify subtle anomalies proposed by the NC model?
- On which factors does the reliability of label feedback depend?

These questions are addressed in the following subsection.

5.5.2 Evaluation of Label Feedback

As mentioned in Subsection 5.5.1, the NC (ED+TI) anomaly detection model and the threshold heuristic were deployed on the visualization and labeling prototype in order to allow for online proposition of potential abnormal signals from sensors OP1, OP2 and OP3. In the following section, the quality both of these anomaly propositions and online label feedback by machine operators are evaluated based on the assumptions made in Section 5.4. Furthermore, annotations obtained during this (first) online label feedback are compared to annotations obtained during a (second) retrospective label feedback where possible.

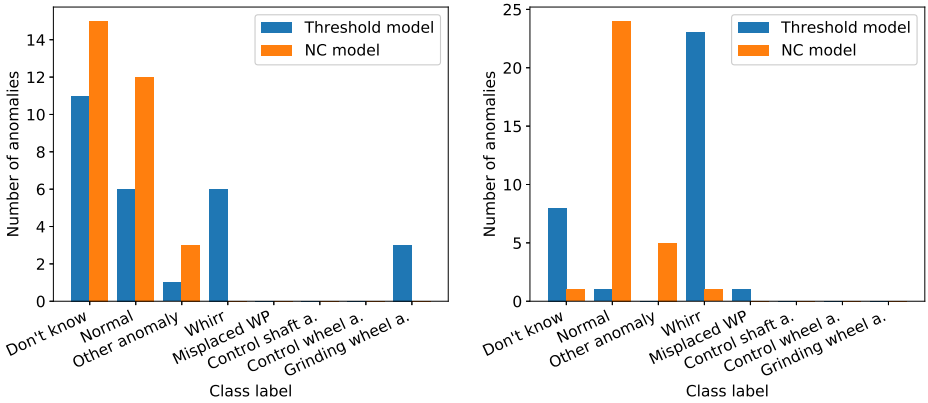
For evaluation of retrospective annotations, access to labels from multiple annotators per each proposed signal as mentioned in Section 5.4 is possible. The machine operators agreed to give these second retrospective annotations for a reasonable amount of signals. The anomaly propositions between 12th and 24th of April were chosen for a second retrospective annotation, as these data comprise the most interesting signals (introduction of the labeling prototype, visually confirmed anomalies around the 16th of April). In order to make the retrospective labels comparable to the single online label, the mode (i.e., majority vote) of retrospective labels is considered in Figures 5.6b, 5.7b and 5.11.

Assumption 1 (Amount and Distribution of Label Feedback)

In Fig. 5.6, the class distribution of anomalies confirmed (true positives) and rejected (false positives) by annotators are stated for both label-proposing algorithms, the NC model and the threshold heuristic. The results are stated both for online label feedback (Fig. 5.6a) and the second retrospective label feedback (Fig. 5.6b). Signals proposed as anomaly but not reacted to during online annotation are thus not displayed in Fig. 5.6a. For retrospective annotation results illustrated in Fig. 5.6b, however, every anomaly proposition was either confirmed, rejected, or labeled with "Don't know" by the annotators.

Considering online annotations in Fig. 5.6a, the threshold heuristic resulted in a smaller degree of false positives than the NC model and less uncertain labels ("Don't know"). Furthermore, clear anomaly types like "Whirr" and "Grinding wheel anomaly" were more frequently identified by the threshold heuristic. Other confirmed anomalies were labeled as unknown types of anomaly ("Other anomalies") and typically identified in reaction to anomaly propositions of the NC model. It is assumed that annotators recognized these data records labeled "Other anomalies" being outliers but were uncertain about the cause and type of these anomalies due to a more subtle deviation across larger parts of the signal than for characteristic "Whirr" and "Grinding wheel anomaly" patterns. This confirms the expectations stated in the last paragraphs of Subsection 5.5.1, that NC models result in a smaller confirmation rate than threshold heuristics, as the latter only propose clearly deviating (high-energy peak) patterns as anomalies but are not capable of considering more subtle signal deviations as anomalies.

For retrospective labeling, different results are observed (cf. Fig. 5.6b): Signals labeled as "Don't know" during online annotation (cf. Fig. 5.6a) were typically labeled either "Normal" or given an anomaly label ("Whirr", "Misplaced work-



(a) *Online labeling (in situ)* ***(b)*** *Retrospective labeling*

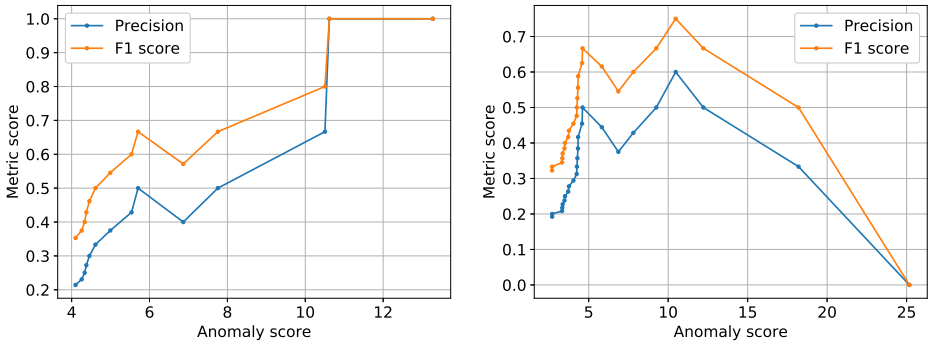
Figure 5.6: *Distribution of annotator feedback across classes (cf. assumption 1): During retrospective annotation (subfigure b), labels are given more confidently to clear classes ("Normal", "Whirr") than during online annotation (subfigure a).*

piece" or "Other anomaly"). It is assumed that the possibility to review signals without time pressure and without the necessity to handle other tasks in parallel encouraged the annotators to take more time during annotation, whereas the daily routine while working at the grinding machine necessitated a more timely reaction to proposed labels. The main difference between online and retrospective annotations was thus found in the redistribution of uncertain labels to more confidence in clearer decisions about the signal being normal or abnormal.

Assumption 2a (Dependency of Label Feedback on Anomaly Scores)

In the former subsection, a high proportion of the online confirmed threshold model anomaly propositions being of clear anomaly types ("Whirr" and "Grinding wheel anomaly") was described. In addition, a dependency of the confirmation rate of NC anomaly propositions on the height of NC anomaly scores as shown in Fig. 5.7 and the time of anomaly proposition as illustrated in Fig. 5.8 was observed.

For anomaly propositions by the NC model, high anomaly scores coincide with high distances between the signal under review and the learned normal centroid. Anomaly scores are thus a measure for the clarity of deviation of a signal under review from the learned normal centroid of the NC model. As more clearly deviating signals proposed as anomalous are assumed to be confirmed an anomaly more frequently, higher accordance between anomaly propositions and label feedback (i.e., both labeled abnormal) is expected for increasing anomaly scores. The confirmation rate is quantified by precision and F1 scores in the following paragraphs.



(a) *Online labeling (in situ)* ***(b)*** *Retrospective labeling*

Figure 5.7: *Dependency of metric scores (precision, F1 score) for label feedback on the height of anomaly scores of the NC model (cf. assumption 2a): For online labels (subfigure a), the dependency between likelihood of confirmation of proposed anomalies and height of anomaly scores is clearer than during retrospective annotation (subfigure b).*

In Fig. 5.7, precision and F1 scores between NC anomaly proposition and label feedback are illustrated across the height of anomaly scores. Precision and F1 scores were computed for binary labels (i.e., all anomaly types are considered a single anomaly class), as the NC model only proposes binary labels (normal vs. abnormal signal). Annotator label feedback was considered as ground truth and anomaly propositions as predicted labels. NC anomaly propositions with label feedback "Don't know" were not considered for computation of the metric scores, as they cannot be assigned either of these binary labels. Anomaly propositions by the threshold model were also not considered in this figure as they come without an intrinsic anomaly score: Neither height nor width nor position of high-amplitude peaks alone seem to be sole reasons for human annotators to confirm a "Whirr" anomaly (as outlined in upcoming evaluations regarding assumption 3b), thus neither of these measures qualifies as anomaly score. NC anomaly detection on the other hand yields a built-in anomaly score based on the distance of test signals to the learned normal centroid, which is additionally related to the visually observable degree of outlierness of a test signal.

The data considered for computation of the metric scores in Fig. 5.7 consists of *(label proposition, label feedback, anomaly score)* triplets. Going from left to right in Figures 5.7a and 5.7b, the triplet with lowest anomaly score from the current set of triplets is successively dropped and F1 score and precision score (between anomaly proposition and label feedback) are computed for the remaining triplets. Thus, the amount of data considered for computation of both metric scores decreases from left to right: While the leftmost plotted point considers all triplets, the rightmost point considers only a single triplet (i.e., the one with the highest anomaly score). For a perfect dependency between the likelihood of confirma-

tion of anomaly propositions and height of NC anomaly score, one would expect a monotonic increase of metric scores from left to right.

For online label feedback, both F1 scores and precision scores increase almost monotonically from left to right and thus with the height of anomaly scores assigned by the NC model. This is interpreted as a confirmation of assumption 2a, that clearer types of anomalies can be detected more reliably by online annotators. For retrospective labeling, a similar dependency of metric scores on the height of NC anomaly scores is observable in Fig. 5.7b when considering a comparable range of anomaly scores as in Fig. 5.7a (anomaly scores between 4 and 13). However, the two rightmost data points in Fig. 5.7b which illustrate the highest anomaly scores were rejected as being normal by retrospective annotators. These rejections effect a sudden decrease in metric scores. The triplets responsible for these two plotted points were not considered in Fig. 5.7a, as they were labeled "Don't know" online and could thus not be judged either as confirmed or rejected anomaly. Thus, a similarly clear dependency between the likelihood of anomaly confirmation and height of NC anomaly scores during retrospective annotation as observed for the online label feedback cannot be found.

Assumption 2b (Dependency of Online Label Feedback on Time)

When illustrating anomaly propositions and online label feedback across time, one observes a temporal dependency of both anomaly propositions and label feedback as illustrated in Fig. 5.8.

Firstly, both anomaly propositions and label feedback cluster at certain days. This is most obviously the case for April 16th and the surrounding days. Annotators confirmed several anomaly propositions with labels "Whirr" and "Grinding wheel anomaly" during online annotation. Visual inspection of the machine validated the annotators' labels: Multiple successive whirring workpieces damaged the grinding wheel and finally resulted in a change of the grinding wheel. Thus, label feedback at these days can be interpreted as reliable. This scenario illustrates an advantage of the live annotation approach: The possibility to consider context information given by the ability to visually inspect the machine during annotation allows for gathering reliable labels of the earliest beginning of grinding machine damages (i.e., the multiple successive whirring workpieces resulting in increasing damages at the grinding wheel surface). This context information cannot be accessed with the common retrospective annotation approaches, where anomalies have to be judged solely relying on the information given by review of sensor signals (as additional information like optical measurements are not available in this study's scenario). As an additional benefit, being able to detect the earliest beginnings of damages in the grinding wheel surface (due to alarm generation for whirring workpieces) allows for the adaptation of process parameters before more severe damages in the grinding wheel damage would necessitate a change of the grinding wheel.

Secondly, an exceptionally high amount of rejected anomaly propositions is observed on the day of introducing the labeling prototype (April 12th), while machine operators never expressed uncertainty about the signal class (label "Don't

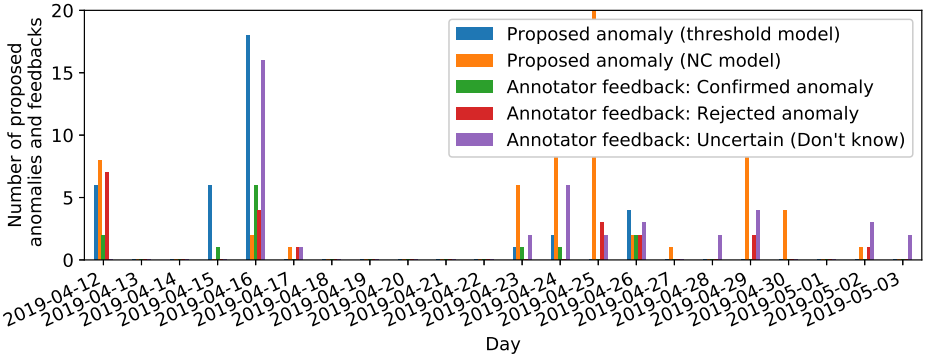


Figure 5.8: *Anomaly propositions and online label feedback across time (cf. assumption 2b). Confirmed anomalies are observed especially around April 16th, where an actual damage of the grinding wheels was confirmed by visual machine inspection.*

know") at all. While the amount of "Don't know" labels increases across time, the amount of label rejections decreases. It is assumed that anomaly rejections were more often replaced by "Don't know" labels due to an increased trust of human annotators in anomaly propositions of the labeling prototype, i.e., small signal deviations were more often rated as potentially abnormal than clearly rejected. Furthermore, the human annotators might have learned new characteristic patterns for signals formerly considered normal due to the anomaly propositions for subtle signal deviations since introduction of the labeling prototype. These effects are considered a "calibration" phase of human annotators having to get accustomed with the labeling prototype before being able to give reliable online label feedback.

Assumption 3a (Inter-Annotator Agreement between Multiple Retrospective Annotators)

In addition to assuming high label reliability for visual clear signal deviations (i.e., high anomaly scores) and days of visually confirmed machine damages, high label reliability was assumed to coincide with a high amount of inter- and intra-annotator agreement in Section 5.4.1. The results both for inter-annotator agreement (among multiple annotators during retrospective labeling) and intra-annotator agreement (between online label feedback and retrospective labels) are illustrated qualitatively for each anomaly proposition of either the NC model or the threshold heuristic in Fig. 5.9. This qualitative evaluation allows judging both class-specific and annotator-specific differences of annotation agreement. Colors encode the class of annotator feedback. Rows 1 to 3 illustrate retrospective labels of multiple annotators. Row 4 depicts the majority vote among these annotators (i.e., mode of rows 1 to 3 per each column). Online label feedback is illustrated in the last row (row 5). Examples of samples with high and low inter-annotator

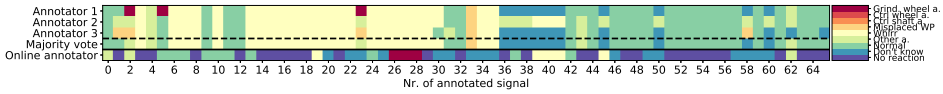


Figure 5.9: *Qualitative evaluation of agreement between multiple retrospective annotators and with online label feedback for signals proposed as anomaly (cf. assumptions 3a and 3b). While annotators show high agreement during retrospective annotation, agreement between in situ online label feedback (row 5) and the majority vote of retrospective annotations (row 4) is low.*

agreement during retrospective labeling are depicted in Fig. 5.10.

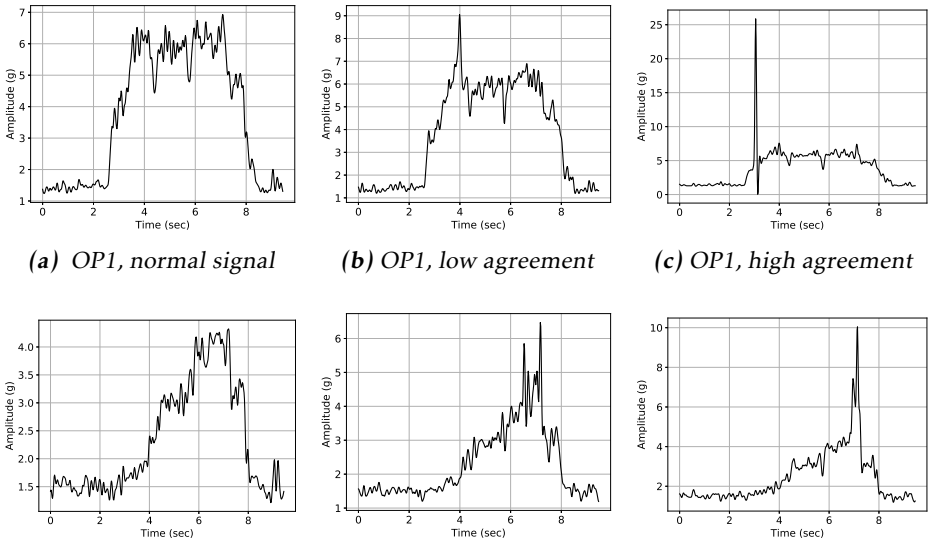
Fig. 5.9 confirms a high inter-annotator agreement during retrospective labeling in general and thus validates interpreting retrospective labels as ground truth labels. The examples with low inter-annotator agreement depicted in Figures 5.10b and 5.10e visually confirm the subtlety of signal deviations in comparison to the depicted normal envelopes in Figures 5.10a and 5.10d. Examples for high inter-annotator agreement as depicted in Figures 5.10c and 5.10f on the other hand illustrate clear anomalous "Whirr" patterns. This confirms the findings for assumption 1 that clear anomalies with well-known characteristics (e.g., whirring workpieces) are identified more reliably.

Assumption 3b (Intra-Annotator Agreement between Online Label Feedback and the Mode of Retrospective Annotations)

Fig. 5.11 summarizes the mismatch between online label feedback and the mode of retrospective labels (i.e., row 4 in Fig. 5.9) as confusion matrix in a multiclass setting. This illustration allows to observe class-specific annotation differences quantitatively, while the annotator-specific information from Fig. 5.9 is lost. As in the above evaluations, high annotation agreement was interpreted to coincide with high annotation reliability.

Similar to the qualitative results reported in Fig. 5.9, a small agreement between online label feedback and retrospective annotations is observed. In addition, the confusion matrix allows to detect class-specific differences in intra-annotator agreement. Signals labeled as "Whirr" during online labeling were confirmed during retrospective labeling or labeled "Don't know". These "Don't know" labels were given for signals with a characteristic high-amplitude peak but at an untypical position (second 9) in the signal. Thus, both a typical position (seconds 3–5 for sensor OP2 and seconds 7–8 for sensor OP1 as depicted in Figures 5.10c and 5.10f) and a certain minimum height of high-amplitude peaks seemed to have been internalized by the operators as necessary conditions to classify a signal as "Whirr".

Signals labeled as "Grinding wheel anomaly" during online annotation were labeled as "Whirr" by all retrospective annotators. This might be due to the fact that grinding wheel damages as observed at the 16th of April typically result from multiple successive whirring workpieces. Thus, a smooth transition be-



(a) OP1, normal signal *(b) OP1, low agreement* *(c) OP1, high agreement*

(d) OP2, normal signal *(e) OP2, low agreement* *(f) OP2, high agreement*

Figure 5.10: *Example signals for high and low inter-annotator disagreement. Examples with high agreement illustrate typical "Whirr" patterns, while examples with low agreement are characterized by more subtle signal deviations.*

tween signal patterns from "Whirr" to "Grinding wheel anomaly" exists. This finding illustrates that context information during (online) annotation was necessary to detect the (visually confirmed) grinding wheel damages.

Signals labeled "Don't know" or "Normal" during online annotation were, in most cases, also given either of these two labels during retrospective annotation or labeled as "Whirr". For these two classes "Don't know" and "Normal", the highest mismatch of online and retrospective labels is observed (i.e., lowest intra-annotator agreement). The high mismatch is understandable for the class "Don't know", which is characterized by a high degree of annotator uncertainty by definition. Reasons for the high mismatch of class "Normal" might be given by a limited visualization of signals on the labeling prototype (i.e., anomaly patters better observable during retrospective annotation) and the necessity to annotate timely during online annotation (resulting in an increased annotation error).

Signals labeled "Other anomaly" during online annotation were either confirmed as "Other anomaly" or rejected as "Normal". One of the signals labeled "Other anomaly" during online label feedback was more clearly specified to be illustrating a wrong type of workpiece being processed by the grinding machine by one of the annotators. As this label was not provided among the class buttons during online annotation (cf. Fig. 5.3c), the annotator labeled it as "Other anomaly" during retrospective labeling but left a note specifying the more de-

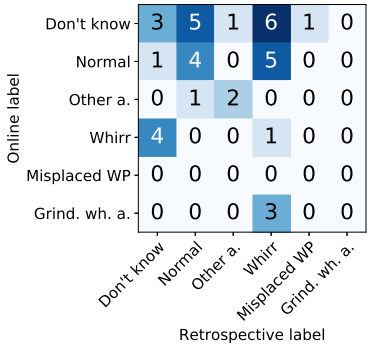


Figure 5.11: *Quantitative comparison of online label feedback and retrospective label feedback (cf. assumption 3b). Similar to the qualitative evaluation in Fig. 5.9, small agreement between online label feedback and retrospective annotations occurs (i.e., most elements are not situated on the main diagonal of the confusion matrix).*

tailed anomaly class specification. This note also specified that the wrong type of workpiece was identified due to a shorter signal length with a characteristic pattern in the end of the signal. Thus, although the signal deviation was small, a characteristic signal pattern could be identified by the retrospective annotator. A personal communication with this annotator confirmed, that having more time during retrospective annotation of this signal was helpful in order to identify the subtle deviation.

In summary, the major findings on label reliability are as follows:

- Dependency of online label feedback on types of anomaly (assumption 1): Clear anomaly types (whirring workpieces, grinding wheel damages) were more often confirmed and typically proposed by the threshold model, whereas subtle anomalies were confirmed more seldom in general and typically proposed by the NC model. This is illustrated in Fig. 5.6.

- Dependency of label feedback on height of anomaly scores (assumption 2a): Higher anomaly scores for anomaly propositions of the NC model resulted in higher precision and F1 scores (cf. Fig. 5.7). This is interpreted to be due to clearer signal deviations that were better observable by the human annotators, resulting in more certain and thus reliable annotations. This dependency was more clearly observable for live annotations than for retrospective annotations.

- Dependency of online label feedback on time (cf. assumption 2b): High amounts both of anomaly propositions and online label feedback clustered at days of visually confirmed machine damages as confirmed by Fig. 5.8. This verifies the sensibility of anomaly propositions at these days and reliability of the high amount of anomaly labels assigned at these days. Furthermore, one observed a “calibration” phase of users getting accustomed with

the labeling prototype, where the labeling behavior of users changed from tending to reject anomaly propositions to reacting with labeling signals as uncertain ("Don't know"). This latter finding is interpreted as increased trust of human annotators in anomaly propositions prompted via the labeling prototype.

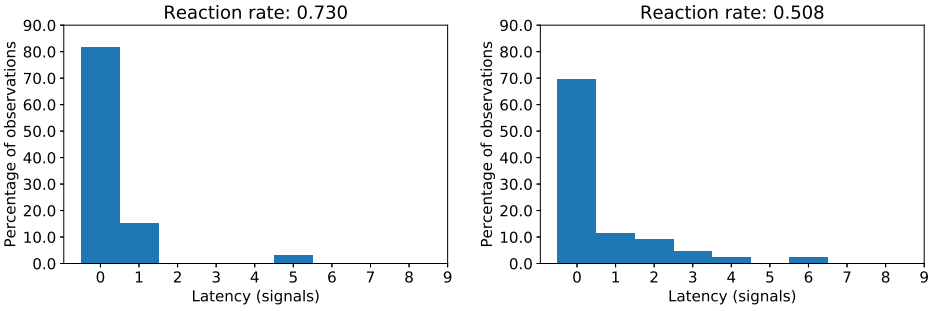
- Reliability of retrospective annotations (assumption 3a): Retrospective annotations illustrated high inter-annotator agreement especially for the class "Whirr" (cf. Fig. 5.9). This confirms high reliability of retrospective labels especially for this "Whirr" class characterized by clearly deviating signals. Furthermore, signal examples illustrated in Fig. 5.10 visually confirm that signals with high inter-annotator agreement were clearly identifiable as signal outliers and depict a typical "Whirr" signal pattern. On the other hand, examples with low inter-annotator agreement were characterized by more subtle deviations.

- Reliability of online annotator feedback (cf. assumption 3b): Similarly, online label feedback showed a high agreement with retrospective labels for the visually clearly identifiable signal deviations of class "Whirr" (cf. Figures 5.9 and 5.11). Subtle and uncertain signal outliers were more likely to be labeled an anomaly during retrospective annotation (cf. Figures 5.6b and 5.9). This clear type of "Whirr" anomalies is thus interpreted to be labeled most reliably during online annotation.

Additional to assumptions on annotation reliability (assumptions 1 to 3) the assumptions on user motivation (assumption 4 to 5) are evaluated.

Assumptions 4a and 4b (Reaction Rate and Reaction Latency during Online Label Feedback)

High user motivation was assumed to coincide with a high reaction rate to anomaly propositions (assumption 4a) and small reaction latencies of feedback to anomaly propositions (assumption 4b). Here, reaction is defined by any feedback by the operator (confirmation, rejection, or label "Don't know"). Fig. 5.12 states reaction rates for both the threshold heuristic and the NC model and illustrates the distribution of observed reaction latencies. Latencies were measured in signals, i.e., a latency of 0 signals represents direct annotator feedback. Reaction rates were measured by the fraction of anomaly propositions which the machine operator reacted to. Both models show a similarly small reaction latency with direct feedback given to most anomaly propositions. For the NC model, a single outlying bin at a reaction latency of 177 signals was omitted due to reasons of visualization of the histogram. These 177 successive NC anomaly propositions with high-latency feedback were prompted on April 23rd and 24th and were characterized by occurring as burst of small anomaly scores (i.e., visually subtle signal deviations). It is assumed that missing feedback for these successive propositions is due to thorough reviewing of subtle signal deviations throughout these episodes of anomaly propositions, i.e., the reviewing spanned multiple of these successive anomaly



(a) Threshold model *(b) NC model*

Figure 5.12: *Reaction rates (cf. assumption 4a) and histograms of reaction latencies (cf. assumption 4b) for online label feedback: Reaction latencies are small for both anomaly detection models. The reaction rate is smaller for the NC model (subfigure b) than for the threshold model (subfigure a).*

propositions. When omitting this single outlying latency value of 177 signals, the NC reaction rate computes to 0.508 (cf. Fig. 5.12b). When considering the outlying latency value, an NC reaction rate of only 0.127 is computed. In both cases, the reaction rate of the NC is smaller compared to the threshold heuristic (0.730, cf. Fig. 5.12a). This is again assumed to be related to the visual clarity of "Whirr" and "Grinding wheel anomaly" patterns in the signals proposed by the threshold heuristic.

Assumption 5 (Dependency of User-initiated Actions on Time)

Finally, Fig. 5.13 illustrates the amount of user-initiated actions during online annotation and its change across time. Similar to Fig. 5.8, one observes a clustering of user-initiated annotations and relearnings close to the visually confirmed grinding wheel damage at April 16th. This is interpreted as a sign of high user motivation, as the amount of user-initiated activity increases when necessary, i.e., for high densities of real anomalies and resulting process adaptations.

In summary, the major findings on user motivation are:

- Relation between user motivation and user reaction latency/rate (assumptions 4a and 4b): Reaction latencies for online label feedback were small for both anomaly proposing models (Fig. 5.12), which is interpreted as a sign of high user motivation. The smaller reaction rate to NC anomaly propositions might be related to the more thorough reviewing of subtle signal deviations which characterized many of the NC anomaly propositions.

- Relation between user motivation and time (assumption 5): User-initiated actions were observed mainly during days of visually confirmed machine damages and resulting machine part changes (i.e., damage and change of

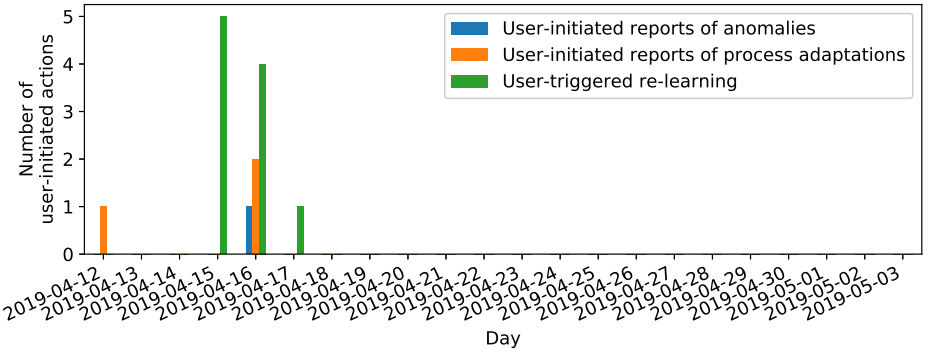


Figure 5.13: *User-initiated actions across time (cf. assumption 5): Reported anomalies, reported process adaptations and user-triggered relearning. Similar to clusters of anomalies in Fig. 5.8, user-initiated actions cluster around April 16th, where damage of the grinding wheels was visually confirmed.*

grinding wheel on April 16th) (Fig. 5.13). This is interpreted as a sign of high user motivation to annotate signals.

5.6 Conclusions

This chapter presented an alternative approach to retrospective annotation of sensor streams in industrial scenarios. Retrospective annotations cause high costs (due to the additional time spent by domain experts for data annotation) and allow only a small amount of context information to be considered during annotation (neither workpieces nor machine tools are accessible for inspection). On the other hand, the proposed live and in situ annotation approach enables highly reduced annotation cost (in-parallel annotation of signals at recording time by domain experts) while exposing a higher amount of meta information for consideration during annotation (possibility to assess both machine tool and workpieces). A drawback of live annotation however is the reduced time for annotation.

The goal of the study presented in this chapter was to examine if and for which types of anomalies live and in situ annotation proves superior to retrospective annotation by the same group of domain experts (machine operators). This was assessed via comparison of live annotations (i.e., machine operator's direct feedback to anomaly propositions) and retrospective annotations (by multiple domain experts) gathered in real-world industrial manufacturing environments. In addition to estimating reliability of live annotations, influential factors on reliability were aimed to be identified. These influential factors were summarized in multiple assumptions and tested on validity with the data collected in this study.

For data collection, a grinding machine in a real-world manufacturing setting was equipped with vibration sensors for long-term measurements. In addition, both hardware and software of a prototypical system for visualization and in situ

annotation of sensor signals were developed. The development process included the design of a suitable GUI for in situ signal annotation, which was guided by end user experience at several steps of the design process. Generic unsupervised anomaly detection algorithms were deployed on the labeling prototype to propose signals for annotation. Operators of the grinding machines reacted to these anomaly propositions with in situ label feedback. This live annotation approach allowed assembling a corpus of 123,942 real-world manufacturing sensor data records with domain expert annotations for various anomaly types.

As expected, a simple threshold heuristic on signal amplitude found the most typical and severe type of anomaly in this study (whirring workpieces) reliably, as amplitude thresholds are tailor-made for its exact type of manifestation in the signals (high-amplitude peaks). Furthermore, anomalies caused by multiple successive whirring workpieces (grinding wheel damages) were detected reliably as confirmed by visual machine inspections. However, many of the signals proposed as anomalous by the threshold model were rejected (FPs) or labeled uncertain regarding the presence of an anomaly (label "Don't know"). This is assumed to be due to operators judging data records as representing whirring workpieces not only dependent on the presence but also a minimum height and expected position of high-amplitude peaks (cf. evaluations in Section 5.5.2, assumption 3b).

The Nearest Centroid (NC) model was deployed as a second anomaly detection model in order to find more subtle types of anomalies with less characteristic patterns than whirring workpieces by means of a sequence-level Euclidean distance measure. A small amount of anomaly propositions was confirmed online with the label "Other anomaly". Most signals proposed as potential anomalies however were labeled as normal (FPs) or uncertain ("Don't know"). The likelihood of a proposed signal to be confirmed as anomaly increased with the height of the NC anomaly score, i.e., the clarity of its signal deviation. All of the above illustrates that it is hard for operators to specify types of subtle anomalies without having internalized a characteristic pattern of manifestation in signals. It is assumed that operators can learn such characteristic patterns over time by being shown multiple examples of these subtle anomalies (as the proposed prototypical system does). However, complementary types of signal representation by TFDs or feature scores might be necessary in order to represent signals in a form where these subtle anomaly types manifest more clearly and in characteristic patterns.

Both the amount of anomaly confirmations and user-initiated actions (reporting anomalies and process adaptations, triggering re-learning of the anomaly detections models after process adaptations) during live annotation clustered with days of visually confirmed machine damages (April 16th), which is interpreted as a sign of reliable labels for the reported anomaly types ("Whirr" and "Grinding wheel anomaly") and good user motivation. The latter was confirmed by small reaction latencies and high reaction rates to online anomaly propositions.

High inter-annotator agreement of multiple annotators during a second, retrospective annotation phase confirmed a higher reliability of annotations for anomaly types with a clear and unique signal pattern: Signals labeled as "Whirr" or "Grinding wheel anomaly" during online annotation were similarly identified as one of these classes during retrospective labeling. Furthermore, being able to in-

spect the grinding machine after the occurrence of whirring workpieces allowed to identify resulting damages in the grinding wheel damages at an early state (i.e., before severe damages require a change of the grinding wheel). It is this context information given by the possibility of machine tool inspection which allows for a reliable annotation of (early) grinding wheel damages in the data. This possibility to inspect the machine tool during emergence of an anomaly is not given during retrospective annotation and verifies the benefit of the presented approach for early identification of and response to these types of clear anomalies.

On the other hand, large differences between retrospective labels and live annotations occurred mainly for subtle anomaly types. This confirms the findings from above that subtle anomaly types are hard to identify without a characteristic internalized pattern of manifestation. For these subtle anomalies, having enough time for an extensive review of signals (as present during retrospective annotation) seems to outweigh the benefit of context information given by inspection of machine tools and workpieces during live and in situ annotation. This was confirmed in discussions with the annotators. Thus, the restricted time for signal review during live annotation was identified as a limiting factor to the proposed live annotation approach when the signals under review illustrated only subtly deviating and unknown, non-characteristic signal patterns.

In summary, the main insight of the study is that anomaly types manifesting in clearly deviating and well-known, characteristic signal patterns can be identified reliably via the proposed live annotation approach. However, other signals proposed as potential anomalies that illustrated an unknown, less characteristic or more subtly deviating signal pattern were mostly rejected, i.e., labeled as normal. The question remains whether the small amount of confirmations of subtle anomalies is caused by insufficient representation of discriminative signal information in envelope signals, the simplicity of the anomaly detection models not being able to detect or even cluster these subtle anomalies or simply seldom occurrences of these types of anomalies in general. This question shall be clarified in the following chapter, where neural network methods are leveraged for a more sophisticated anomaly detection. In detail, the focus in the next chapter is on:

- Finding advanced anomaly detection models which learn a more descriptive and reliable latent embedding representation of the normal class data. The goal of this is reducing the FP rate of the anomaly proposing model in the live annotation approach.
- In addition, semi-supervised and weakly supervised extensions of these deep anomaly detection models are evaluated in order to clarify whether including labels allows to better align anomaly propositions with the operator's concept of what an anomaly is.

5.7 Related Publications

C. Reich, A. Mansour, and K. Van Laerhoven. Collecting labels for rare anomalies via direct human feedback — an industrial application study. *Informatics*, 6(3):article nr. 38, September 2019.

6

Neural Anomaly Detection

This chapter presents a comparison of neural anomaly detection models defined by various combinations of neural network encoders and anomaly detection related loss functions. Anomaly detection in this chapter is restricted to binary classification scenarios, i.e., it assumes two classes of normal and abnormal data. The main goal is finding a more powerful model for proposition of potential anomalies than the models presented in the previous Chapter 5 (nearest centroid models and threshold heuristics). For this purpose, the neural anomaly detection models presented in this chapter need to learn a more descriptive time series representation of the sensor data.

First, Section 6.2 outlines loss functions which are popular in neural anomaly detection applications. Then, various hidden layer types which have proven powerful across a wide range of time series representation learning tasks [24, 266] are discussed. From these layer types, various encoder networks are constructed and combined with the loss function types presented before. Afterwards, details on training and optimizing these encoder-loss function combinations are outlined. Finally, an existing approach for automatic generation of labels from weak information sources [21] is discussed.

Then, experiments for evaluation of these encoder-loss function combinations are performed in Section 6.3 regarding their capability to learn a sensible representation of the normal data and to detect outliers from these normal data (i.e., anomalies). After description of the evaluation data (consisting of several days with in-the-wild recorded anomalies) and definition of a generic model size for all encoder types, visualizations of the learned time series representations and anomaly scores predicted with the learned encoders are presented for various unsupervised model combinations. These qualitative results are complemented by a quantitative evaluation of these models via performance metrics. Afterwards, semi-supervised and weakly supervised extensions of these models are compared

via the same visualizations and metrics. The weakly supervised model utilizes automatically generated labels as mentioned in the previous paragraph. Finally, the unsupervised, semi-supervised and weakly supervised models are benchmarked on the live annotation data collected in the previous chapter regarding their capability to more reliably present potential anomalies for annotation (i.e., reduce the high FP rate observed for the simple anomaly detection models in Chapter 5).

The weakly supervised model extension is trained via a novel loss function custom-built for anomaly detection from weakly labeled data. Both this loss function and a subset of the methods and experiments presented in this chapter were first presented in the master's thesis [110] supervised by the author.

6.1 Motivation

Among the most interesting findings from the user study conducted in Chapter 5 is the large disagreement between recorded data proposed as abnormal by the applied anomaly detection models and the feedback to these propositions both by live annotators and retrospective annotators. This high disagreement was visualized in Fig. 5.9. The high disagreement suggests the operators not recognizing the subtlety of signal deviations or not knowing the type of the proposed anomalies. Alternatively, the high FP rate might be caused by the simple applied anomaly detection models not being capable of capturing the complex normal behavior of the evaluated grinding machine from which the data was recorded. The high agreement among retrospective annotators suggested the latter explanation to be more likely: The applied nearest centroid (NC) anomaly detector sacrifices historical knowledge about normal behavior incorporated in previously observed data for having an always up-to-date representation of the normal data (captured by the iteratively updated normal centroid). Thus, the NC model is not capable of storing historical information about the machine's normal behavior given by the vast amount of previously analyzed data in a memory-efficient way.

This chapter compares various alternative anomaly detection models with the goal of finding a sophisticated model being able to better represent the normal behavior of the analyzed grinding machine than the simple NC models, ultimately reducing the high FP rate observed for these simple models. Several unsupervised models, defined by various combinations of neural time series encoder types and popular anomaly detection loss functions, are compared on a large set of sensor data recorded from the same grinding machine as in the previous chapter. This large data set illustrates a high degree of covariate shift of normal data, i.e., the manifestation of normal behavior in the data evolves throughout the successive days of recording. This covariate shift is caused both by process adaptations and (e.g., temperature-related) drift. Keeping an up-to-date representation of the normal data despite the large covariate shift of the data constitutes the biggest challenge for the compared models. In addition, the models have to obey constraints imposed by the embedded nature of the evaluation system (restricted memory space) and application (timely responses to potential anomalies necessitate short model execution times).

In addition to these unsupervised models, semi-supervised extensions provided with a small fraction of expert labels are compared to the purely unsupervised anomaly detectors in order to evaluate the benefit of including label information. Finally, a weakly supervised extension provided with automatically generated labels and trained with a novel anomaly detection loss function is evaluated. These semi-supervised and weakly supervised model extensions prove to be superior both to simple anomaly detection models applied in the previous Chapter 5 and the purely unsupervised models evaluated in this chapter.

6.2 Methods

This section outlines details of the main building blocks for defining neural anomaly detection models. These are subdivided into loss functions as outlined in Subsection 6.2.1 and network layers as discussed in Subsection 6.2.2. Network layers in turn are used to define various types of encoders and decoders for creating several neural anomaly detection models. Encoders and decoders are described in Subsections B.1 and B.2, respectively. Finally, a variety of encoder-decoder architectures is presented in Subsection 6.2.3, accompanied by an explanation of routines applied for training these models and performing hyperparameter optimization.

In addition, methods for automatic generation of labels are discussed in Subsection 6.2.4. For this, PGMs are defined which allow estimating a final reliable label estimate from automated weak labeling functions.

6.2.1 Loss Functions

There are various loss functions applied in the upcoming experiments and in neural anomaly detection applications in general. Autoencoder [101] and variational autoencoder [124] loss functions represent the most frequently chosen types of loss functions in anomaly detection applications [207]. These loss functions judge the degree of a data record to be anomalous by the reconstruction error and reconstruction probability, retrospectively. Although both are reasonable assumptions for anomalous behavior, one-class loss functions like the Deep SVDD loss [207] represent a more direct way of modeling the central idea inherent to most anomaly detection tasks [7]: Abnormal data manifest as outliers from the majority of data. The following section thus puts an emphasis on different formulations of the Deep SVDD loss and a semi-supervised extension of these. Finally, a novel weakly supervised extension of the Deep SVDD loss function is proposed.

Autoencoder

Autoencoders (AEs) are among most popular representation learning methods and a typical starting point for anomaly detection in various application domains [207]. AEs consist of an encoder f and a decoder g, both represented by neural networks. Predicting anomalies via AEs is performed by interpreting

the reconstruction error $\|\mathbf{x} - \hat{\mathbf{x}}\|_2$ between a data record $\mathbf{x}$ and its reconstruction $\hat{\mathbf{x}} = g(f(\mathbf{x}))$ as an anomaly score. The (mean) reconstruction error is also used during training of AEs, where the following loss function is minimized for the training data $\mathcal{D}_N = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$:

$$\mathcal{L}(\mathcal{W}; \mathbf{x}_i) = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{x_i} - g(f(\mathbf{x_i}))\|_2^2 + \frac{\lambda}{2} \sum_{l=1}^{L} \|\mathbf{W}_l\|_F^2 \tag{6.1}$$

Here, $\mathcal{W} = \{\mathbf{W}_1, \ldots, \mathbf{W}_L\}$ are the weights $\mathbf{W}_l$ of layers l of the network. The autoencoder loss function is defined as a regularized optimization problem, with the first term being the mean reconstruction error of the training data records $\mathbf{x}_i \in \mathcal{D}_N$ and the second term a weight regularizer which penalizes overfitting of the network weights to the training data. λ is a hyperparameter for trading off the influence of both terms, $\|\cdot\|_F$ indicates the Frobenius norm computed for the weights $\mathbf{W}_l$. The encoder f is typically enforced to learn a compact intermediate representation $\mathbf{p}$ of the data by imposing a compression factor $1 \leq c_f = \frac{d}{p} \leq d$ defined by the input layer dimension d and the dimension p of the intermediate representation $\mathbf{p}$.

Thus, the optimal AE model is found by minimizing $\mathcal{L}(\mathcal{W}; \mathbf{x}_i)$ denoted as regularized mean reconstruction error across the training data. Minimizing this AE loss function can result in a compact representation of the data but does not capture the generative process of the data. This in turn can result in a poor generalization performance of the learned AE model to unseen test data. Such poor generalization applies especially in domain adaptation scenarios, where a non-neglectable difference between the distributions of training data and test data is assumed [104, 105]. One of the main types of domain adaptation scenarios is the necessity to handle covariate shift of the data. As mentioned in the beginning of this chapter, large degrees of covariate shift represent the main challenge for the representation learners compared in this study, thus making this potentially poor generalization performance of AEs an actual problem. The drawback of potentially poor generalization performance is addressed by deep generative models. Among deep generative models, variational autoencoders (VAEs) are one of the most popular model families.

Variational Autoencoder

Other than traditional AE approaches, VAEs do not aim at finding a compact intermediate representation of the data but to model the data-generating distribution $p(\mathbf{x})$ [124]. If the data records $\mathbf{x}$ are mainly from the normal data class, anomaly detection can effectively be framed as the problem of estimating the probability density function of the data-generating distribution $p(\mathbf{x})$: The probability of abnormal records under this distribution is low.

Often, directly estimating the density of $p(\mathbf{x})$ by marginal likelihood maximization is hard due the complexity and high dimensionality of the data $\mathbf{x}$. VAEs circumvent the challenge of directly estimating $p(\mathbf{x})$ by introducing latent variables $\mathbf{z}$. The basic idea is to combine a latent distribution $p(\mathbf{z})$ which can be easily

sampled from (like a normal distribution) with a nonlinear decoder function $g(\mathbf{z})$ which allows to produce samples $\hat{\mathbf{x}}$ closely resembling the true records $\mathbf{x}$ by passing samples $\mathbf{z}$ through $g(\mathbf{z})$. The complexity of the decoder function $g(\mathbf{z})$ which is necessary to mimic the complexity of $p(\mathbf{x})$ can be represented by a neural network $g(\mathbf{z};\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ are learnable parameters of the network.

Estimating the density of $p(\mathbf{x})$ can then be interpreted as leveraging Bayes' rule $p(\mathbf{x}) = \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$. Here, $g(\mathbf{z};\boldsymbol{\theta})$ was replaced by the distribution $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ which allows to explicitly formulate the dependency of $\mathbf{x}$ on $\mathbf{z}$ [68]. The distribution $p(\mathbf{x})$ can be approximated by an empirical expectation for samples $\mathbf{z}_i$, i.e., $p(\mathbf{x}) = \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} \approx \frac{1}{m}\sum_i p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_i)$. For complicated distributions $p(\mathbf{x})$, a large number of samples $\mathbf{z}_i$ might be necessary to approximate $p(\mathbf{x})$ as many of the sample probabilities $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_i)$ might be close to zero [68]. Thus, finding a good proposal distribution $\mathbf{z}_i \sim q(\mathbf{z})$ that minimizes the number of samples $\mathbf{z}_i$ necessary for approximating $p(\mathbf{x})$ is crucial. For this, the proposal distribution $q(\mathbf{z})$ should approximate $p(\mathbf{z})$ as closely as possible in order to generate such samples $\mathbf{z_i}$ with $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_i) >> 0$. VAEs address this challenge by leveraging variational methods for learning the proposal distribution $q(\mathbf{z})$ directly from the data $\mathbf{x}$ with a second neural network $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$. This second network with learnable parameters $\boldsymbol{\phi}$ can be interpreted as an encoder network.

Then, learning optimal parameters for both networks $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ and $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ can be used to approximately maximize the intractable marginal likelihood $p(\mathbf{x})$. For this purpose, VAEs leverage variational methods by maximizing the evidence lower bound (ELBO) on the log marginal likelihood $\log p(\mathbf{x})$ (also referred to as log *evidence*). The ELBO is represented by the right-hand side of the following inequality:

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] - D_{KL}[q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})] \tag{6.2}$$

Here, $\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})]$ denotes the empirical approximation of $p(\mathbf{x})$ resembling the form of an autoencoder, as $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ encodes $\mathbf{x}$ into $\mathbf{z}$ and $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ decodes $\mathbf{x}$ from $\mathbf{z}$ [68]. This allows interpreting $\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})]$ as the expected negative reconstruction error [124]. The second term denotes a distance measure between $p(\mathbf{z})$ and the proposal distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ that aims approximating $p(\mathbf{z})$. The distance between both distributions is measured by the Kullback-Leibler divergence D_{KL}. For a high-capacity neural encoder $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$, this distance can be brought close to zero. Thus, by minimizing this second term, the empirical approximation in the first term is allowed to approach the marginal likelihood $p(\mathbf{x})$ on the left-hand side of Eq. 6.2 as close as possible [68].

In summary, VAEs replace the problem of estimating the density of $p(\mathbf{x})$ by an optimization problem, where learning an encoder $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ minimizing the divergence $\mathcal{D}_{KL}[q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})]$ and a decoder $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ maximizing the empirical expectation $\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})]$ allow to maximize the variational lower bound on $p(\mathbf{x})$:

$$\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x}_i) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)}[\log p_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z})] - D_{KL}[q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_i)\|p_{\boldsymbol{\theta}}(\mathbf{z})] \tag{6.3}$$

The loss function defined in Eq. 6.3 can be optimized via stochastic gradient descent (SGD) for the training data $\mathcal{D}_N = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. In order to make the training

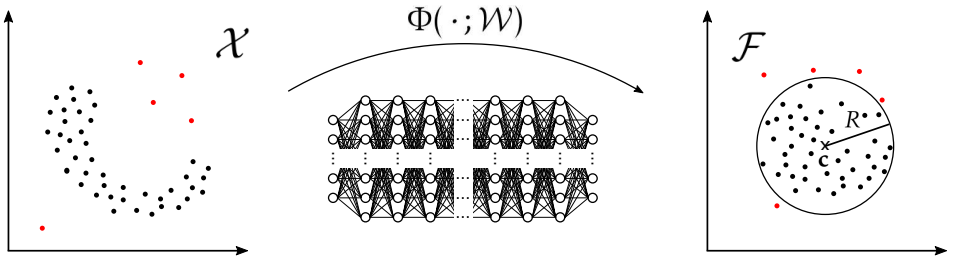


Figure 6.1: *Outline of the Deep SVDD approach. Deep SVDD aims to learn a transform $\Phi(\cdot;\mathcal{W})$ from input space $\mathcal{X}$ to output space $\mathcal{F}$ that maps the majority of the data (black dots) into an enclosing hypersphere with minimum volume. The hypersphere is defined by its center $\mathbf{c}$ and a radius R. Outliers (red dots) of this majority of (normal) data are separated by the enclosing hypersphere. Figure inspired from [207].*

procedure work, different adaptations of the loss function described in Eq. 6.3 have to be made during implementation. Most importantly, the non-continuous sampling operation $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$ has to be moved to an input layer in order to be able to compute the gradients of both networks during backpropagation. This technique is referred to as *reparameterization trick* [124]. For a detailed discussion, the interested reader is referred to [68].

Deep Support Vector Data Description (Deep SVDD)

Although AEs and VAEs are among most popular applied anomaly detection losses, both loss functions were originally proposed with other goals in mind. In [207], Ruff et al. proposed the Deep SVDD loss function being specifically tailored to anomaly detection applications.

Deep SVDDs try to find a mapping $\Phi(\cdot;\mathcal{W})$ from the input data space $\mathcal{X} \subseteq \mathbb{R}^d$ to an output space $\mathcal{F} \subseteq \mathbb{R}^p$ such that the majority of the mapped data is enclosed by a hypersphere of minimum volume. Here, the output space represents the embedding space for the time series representations that shall be learned. The hypersphere is parameterized by its center $\mathbf{c} \in \mathcal{F}$ and a radius R. The volume of the hypersphere is thus minimized by minimizing R. This is summarized in Fig. 6.1.

The mapping $\Phi(\cdot;\mathcal{W})$ can be parameterized by a neural network. In order to learn the parameters $\mathcal{W} = \{\mathbf{W}_1, \ldots, \mathbf{W}_L\}$ for L network layers, the following loss function is defined and optimized for the training data $\mathcal{D}_N = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ [207]:

$$\mathcal{L}(R, \mathcal{W}; \mathbf{x}_i) = R^2 + \frac{1}{\nu N} \sum_{i=1}^{N} \max\{0, \|\phi(\mathbf{x}_i; \mathcal{W}) - \mathbf{c}\|_2^2 - R^2\} + \frac{\lambda}{2} \sum_{l=1}^{L} \|\mathbf{W}_l\|_F^2 \qquad (6.4)$$

This equation is referred to as *soft-boundary Deep SVDD* loss function. The problem of minimizing R (first summand) is penalized by two other terms: A penalty

term for data points $\phi(\mathbf{x}_i; \mathcal{W})$ situated outside the hypersphere (second term) and the same weight regularizer (third term) as for the AE loss function in Eq. 6.1. The second term penalizes data points depending on their distance $\|\phi(\mathbf{x}_i; \mathcal{W}) - \mathbf{c}\|$ to the hypersphere center $\mathbf{c}$ if this distance is greater than the hypersphere radius R. The hyperparameter $\nu \in (0, 1]$ trades off the influence of the first two terms, i.e., the volume of the hypersphere and violations of the SVDD boundary, thus allowing some points to be mapped outside the hypersphere. This is similar to the concept of slack variables in the soft-boundary loss function of classical SVMs [40], thus the term *soft-boundary SVDD* loss function.

The center of the hypersphere $\mathbf{c}$ is not optimized via Eq. 6.4. This hyperparameter $\mathbf{c}$ is assumed to be fixed and can be computed as the average of all data records $\mathbf{x}_i$ mapped to the output space $\mathcal{F}$ after a forward pass through the network. Thus, a sensible initialization of the network weights $\mathcal{W}$ is crucial in order to obtain a reliable estimate of $\mathbf{c}$ via this initial forward pass. In [207], this weight initialization is performed by pre-training the Deep SVDD network with a variant of the AE loss function.

The soft-boundary Deep SVDD objective as defined in Eq. 6.4 can be simplified when the majority of the training data $\mathcal{D}_N$ is considered to be of one class. Then, the following objective can be defined [207]:

$$\mathcal{L}(\mathcal{W}; \mathbf{x}_i) = \frac{1}{N} \sum_{i=1}^{N} \|\phi(\mathbf{x}_i; \mathcal{W}) - \mathbf{c}\|_2^2 + \frac{\lambda}{2} \sum_{l=1}^{L} \|\mathbf{W}_l\|_F^2 \tag{6.5}$$

This objective is referred to as *one-class Deep SVDD* loss function. Its minimum is found when all data points are mapped as close as possible to the center $\mathbf{c}$ of the hypersphere. This approach of trying to map all the data close to the center $\mathbf{c}$ is valid for training data consisting mainly of data from a single class. This assumption in turn is justified in anomaly detection scenarios.

Semi-Supervised Deep SVDD

In [208], Ruff et al. extend their (unsupervised) Deep SVDD approach [207] to a semi-supervised loss function. They refer to this extension as *Deep Semi-Supervised Anomaly Detection (SAD)*. Here, the training data is assumed to consist of unlabeled data $\mathcal{D}_N = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and labeled data $\mathcal{D}_M = \{(\tilde{\mathbf{x}}_1, \tilde{y}_1), \ldots, (\tilde{\mathbf{x}}_M, \tilde{y}_M)\}$. The semi-supervised loss function is then defined as follows:

$$\mathcal{L}(\mathcal{W}; \mathbf{x}_i) = \frac{1}{N+M} \sum_{i=1}^{N} \|\phi(\mathbf{x}_i; \mathcal{W}) - \mathbf{c}\|_2^2 + \frac{\eta}{N+M} \sum_{j=1}^{M} (\|\phi(\tilde{\mathbf{x}}_\mathbf{j}; \mathcal{W}) - \mathbf{c}\|_2^2)^{\tilde{y}_j} + \frac{\lambda}{2} \sum_{l=1}^{L} \|\mathbf{W}_l\|_F^2 \tag{6.6}$$

This loss function consists of three terms. The first and third term are similar to the one-class Deep SVDD loss defined in Eq. 6.5. The second term considers information given by the labels $\tilde{y}_j \in \{+1, -1\}$, where $\tilde{y}_j = +1$ indicates a normal data record and $\tilde{y}_j = -1$ an abnormal data record. The label information is used to minimize the distance of normal training data records $\tilde{\mathbf{x}}_j = 1 \in \mathcal{D}_M$

to the hypersphere center $\mathbf{c}$ while maximizing the distance of abnormal training data records $\tilde{\mathbf{x}}_j = -1 \in \mathcal{D}_M$ to this center. Maximizing the distance of abnormal training data records to the hypersphere center involves inversion of the term $(\|\phi(\tilde{\mathbf{x}}_j; \mathcal{W}) - \mathbf{c}\|_2^2)^{\tilde{y}_j}$ due to $\tilde{y}_j = -1$. For numerical stability, the authors of [208] add an epsilon $\texttt{eps} \approx 10^{-6}$ to the denominator of this inverse. The second term of Eq. 6.6 is weighted by a hyperparameter η, which governs the influence of the unlabeled training data $\mathcal{D}_N$ and the labeled training data $\mathcal{D}_M$. For $M = 0$, the semi-supervised Deep SVDD loss function reduces to the one-class Deep SVDD loss function defined in Eq. 6.5.

Weakly Supervised Deep SVDD

In this paragraph, a weakly supervised extension of the semi-supervised Deep SVDD loss function defined in Eq. 6.6 is described. This weakly supervised Deep SVDD loss function was originally proposed in the master's thesis [110], which was supervised by the author of this doctoral thesis. The extension allows to consider label uncertainty for the training data records $\mathcal{D}_M$. Here, label uncertainty can be quantified by probabilities $p(\tilde{y}_j) \in [0, 1]$. These probabilities are estimated by the probabilistic graphical model (PGM) described at the very end of this section.

Let probabilities $p(\tilde{y}_j)$ be a measure for the certainty of a given class label $\tilde{y}_j \in +1, -1$, i.e., $p(\tilde{y}_j) = 1$ indicating absolute certainty of the label being correct. Now, consider the second term in the semi-supervised Deep SVDD loss function in Eq. 6.6, which comprises the information of the labeled training subset $\mathcal{D}_M$:

$$\star = \frac{\eta}{N + M} \sum_{j=1}^{M} (\|\phi(\tilde{\mathbf{x}}_j; \mathcal{W}) - \mathbf{c}\|_2^2)^{\tilde{y}_j} \tag{6.7}$$

Not considering label uncertainties can then be interpreted as an implicit assumption of all labels being correct, i.e., $p(\tilde{y}_j) = 1 \ \forall \ \tilde{\mathbf{x}}_j \in \mathcal{D}_M$. This assumption can be made explicit by adding the label uncertainty estimates $p(\tilde{y}_j)$ in Eq. 6.7:

$$\star = \frac{\eta}{N + M} \sum_{j=1}^{M} p(\tilde{y}_j) \cdot (\|\phi(\tilde{\mathbf{x}}_j; \mathcal{W}) - \mathbf{c}\|_2^2)^{\tilde{y}_j} \tag{6.8}$$

Assuming absolute certainty of the labels, i.e., probabilities $p(\tilde{y}_j) = 1 \ \forall \ \tilde{\mathbf{x}}_j \in \mathcal{D}_M$, results in an implicit sum of M of the individual weights $p(\tilde{y}_j)$ in Eq. 6.8:

$$\sum_{j=1}^{M} p(\tilde{y}_j) = M \tag{6.9}$$

When allowing uncertainty in the labels by $p(\tilde{y}_j) \in [0, 1]$, this sum reduces, implicitly resulting in a lower weighting of the labeled loss term in Eq. 6.6 than for the case of absolute label certainty (semi-supervised scenario). In order to

recover the same weighting, a constant correction factor c_M can be introduced:

$$c_M = \frac{M}{\sum_{j=1}^{M} p(\tilde{y}_j)} \tag{6.10}$$

For absolute certainty of all labels $p(\tilde{y}_j) = 1 \;\forall\; \tilde{\mathbf{x}}_j \in \mathcal{D}_M$ (semi-supervised scenario), the same weighting as in Eq. 6.6 is recovered: $\sum_{j=1}^{M} p(\tilde{y}_j) = M$, thus $c_M = 1$. For the assumption of weak labels $p(\tilde{y}_j) \in [0, 1] \;\forall\; \tilde{\mathbf{x}}_j \in \mathcal{D}_M$ however, the factor c_M corrects for the implicit smaller weighting of the labeled data term in Eq. 6.7 due to the probabilities $p(\tilde{y}_j)$ inside the sum of Eq. 6.8: The sum of probabilities $\sum_{j=1}^{M} p(\tilde{y}_j) < M$ of weak labels is smaller than for strong labels $\sum_{j=1}^{M} p(\tilde{y}_j) = M$. If the influence of the second, labeled data term in Eq. 6.6 shall further only be governed by the value specified for η, a factor c_M allows correcting for the smaller summed weights of the weak label probabilities $p(\tilde{y}_j)$. Considering the correction factor c_M, the labeled loss term in Eq. 6.7 can then be rewritten as

$$\star = \frac{c_M \eta}{N + M} \sum_{j=1}^{M} p(\tilde{y}_j) \cdot (\|\phi(\tilde{\mathbf{x}}_j; \mathcal{W}) - \mathbf{c}\|_2^2)^{\tilde{y}_j} \tag{6.11}$$

A weakly supervised Deep SVDD loss function can then be proposed as follows:

$$\begin{aligned} \mathcal{L}(\mathcal{W}; \mathbf{x}_i) = \frac{1}{N + M} \sum_{i=1}^{N} \|\phi(\mathbf{x}_i; \mathcal{W}) - \mathbf{c}\|_2^2 + \frac{c_M \eta}{N + M} \sum_{j=1}^{M} p(\tilde{y}_j) \cdot (\|\phi(\tilde{\mathbf{x}}_j; \mathcal{W}) - \mathbf{c}\|_2^2)^{\tilde{y}_j} \\ + \frac{\lambda}{2} \sum_{l=1}^{L} \|\mathbf{W}_l\|_F^2 \end{aligned} \tag{6.12}$$

In summary, defining the correction factor c_M as proposed above allows trading off the mutual influence of the two first terms of Eq. 6.12 solely based on the factor η for all cases of label uncertainty $p(\tilde{y}_j)$. Most importantly, the semi-supervised scenario of $p(\tilde{y}_j) = 1 \;\forall\; \tilde{\mathbf{x}}_j \in \mathcal{D}_M$ can be sensibly compared to weakly supervised scenarios.

The weakly supervised Deep SVDD loss function allows considering individual uncertainty estimates $p(\tilde{y}_j)$ accompanying labels $\tilde{y}_j$ for each data record j. In Subsection 6.3.3, benefits of incorporating these per-record uncertainty estimates are illustrated for a neural anomaly detection model trained with the loss function defined in Eq. 6.12. While label noise aware approaches have been defined for shallow and deep classifiers [196] in class-balanced scenarios before, training models with a label noise aware loss function tailor-made for anomaly detection applications as proposed in Eq. 6.12 is novel to best of the author's knowledge.

6.2.2 Network Layers

Aside from the choice of loss functions, the architecture of a neural network heavily influences its learned internal time series representation. This is mainly due

to the learning bias introduced by the type of chosen network layers. Thus, the choice of network layers as building blocks of each neural network architecture heavily influences performance and generalization of the trained network on new and unseen test data. In the following subsections, input layers and a variety of hidden layers as applied in the upcoming experiments are discussed.

A descriptive summary of the notation used for describing network layers and encoder-decoder architectures built from these layers is given in Table 6.1.

Input Layer

The most straightforward input to network architectures is using the raw sensor data records of (variable) length T_{in}. For sequential neural models, the performance of models directly depends on the history of input data (i.e., information given by previously seen input time series) they are able to store and consider for prediction of incoming data. Architectures based on recurrent neural networks are effective in storing this history of previous input data: It is summarized and stored in hidden states which are kept through time, while original time series can be discarded. Sequential neural models building on (dilated) convolutions however (which are in focus of this chapter) have a history being directly influenced by the length of stored input data: Predictions for newly incoming time series rely on applying convolutional filters directly to stored original input time series. Thus, the history that convolutional models are able to consider during prediction (also referred to as their *receptive field*) by applying a set of convolutional filters depends on the amount of previously stored input time series.

The number and length of stored time series thus directly influence required memory resources and prediction time with the model. For high-performant convolutional models with a large receptive field, the necessary memory occupation can quickly exceed the small memory budget given by the embedded nature of the evaluation system. In order to milden memory requirements while still keeping a reasonably large history, downsampling can be applied to raw input time series. This reduces the actual memory size occupied by storage of (downsampled) time series.

Downsampling has previously been approached by a combination of dilated convolutions and skip connections for autoregressive networks [255]. This results however in a large number of additional learnable parameters. Alternatively, deterministic downsampling operations can be applied. For comparability with previous chapters, downsampling is implicitly performed by extraction of average rectified values (ARVs). ARVs are computed by the mean of absolute values in non-overlapping fixed blocks of $M = 1024$ raw samples u_m, i.e., $\frac{1}{M}\sum_{m=1}^{M}|u_m|$ in every successive signal block comprising M raw data samples u_m. Similar to previous chapters, these ARV representations are referred to as envelope signals in the following.

ARV operations are performed on resampled and offset-freed raw time series. Resampling to a fixed length T_{in} is applied to obtain fixed-length time series necessitated by some of the encoders presented below. Offset subtraction is applied in order to reduce the influence of spurious (e.g., temperature-related) drifts.

Table 6.1: *Notations used for description of network architectures*

Notation	Meaning	Notation	Meaning
$\mathbf{x}$	Input vector	N_c	Number of channels
$\mathbf{y}$	Output vector	N_f	Number of filters
T	Signal length	k	Kernel size
$\mathbf{p}$	Embedding vector	pd	Padding size
p	Embedding dimension	s	Stride
B	Batch size	d	Dilation factor

Hidden Layer Types

Envelope extraction via ARV can be interpreted as input layer. Envelope signals as extracted by this input layer are then fed to hidden layers. A descriptive summary of the hidden layer types considered in this chapter is illustrated in Fig. 6.2. For all layer types, batch normalization [109] and dropout [238] (during training) are optional but common extensions being applied to the network layers. For all network layers, leaky rectified linear unit (ReLU) activation functions are used [152]. Bias terms of network layers are dropped in order to satisfy the requirements imposed by loss functions building on the Deep SVDD loss [207].

Multilayer Perceptron (MLP) Layer MLP hidden layers are fully connected to input vectors $\mathbf{x} \in \mathbb{R}^{T_{in}}$ and output vectors $\mathbf{y} \in \mathbb{R}^{T_{out}}$. Thus, the layer's transform from an input vector $\mathbf{x}$ to an output vector $\mathbf{y}$ can be expressed by a weight matrix $\mathbf{W} \in \mathbb{R}^{T_{in} \times T_{out}}$, followed by batch normalization, passing through the nonlinear Leaky ReLU activation function and dropout.

Convolutional Layer Other than for MLP layers, hidden 1D convolutional layers transform input tensors $\mathbf{X} \in \mathbb{R}^{T_{in} \times N_f^{in}}$ to output tensors $\mathbf{Y} \in \mathbb{R}^{T_{out} \times N_f^{out}}$ leveraging a 1D convolution followed by the same steps of batch normalization, Leaky ReLU and dropout. Here, output tensors $\mathbf{Y}$ consist of output vectors $\mathbf{y}$ for N_f^{out} output channels. The free parameters of convolutional layers determining the output sequence length T_{out} are kernel size k, padding size pd and stride s. The number of output channels N_f^{out} depends on the number of convolutional filters applied in the layer. This is another free parameter of the network layer.

Temporal Convolutional Network (TCN) Residual Block TCN residual blocks are the basic building elements of TCNs [24]. Similar to hidden convolutional layers, they transform input tensors $\mathbf{X} \in \mathbb{R}^{T \times N_f^{in}}$ to output tensors $\mathbf{Y} \in \mathbb{R}^{T \times N_f^{out}}$. They consist of mainly two parts: A stack of dilated causal convolutional filters and residual connections. These main elements are illustrated in Fig. 6.3.

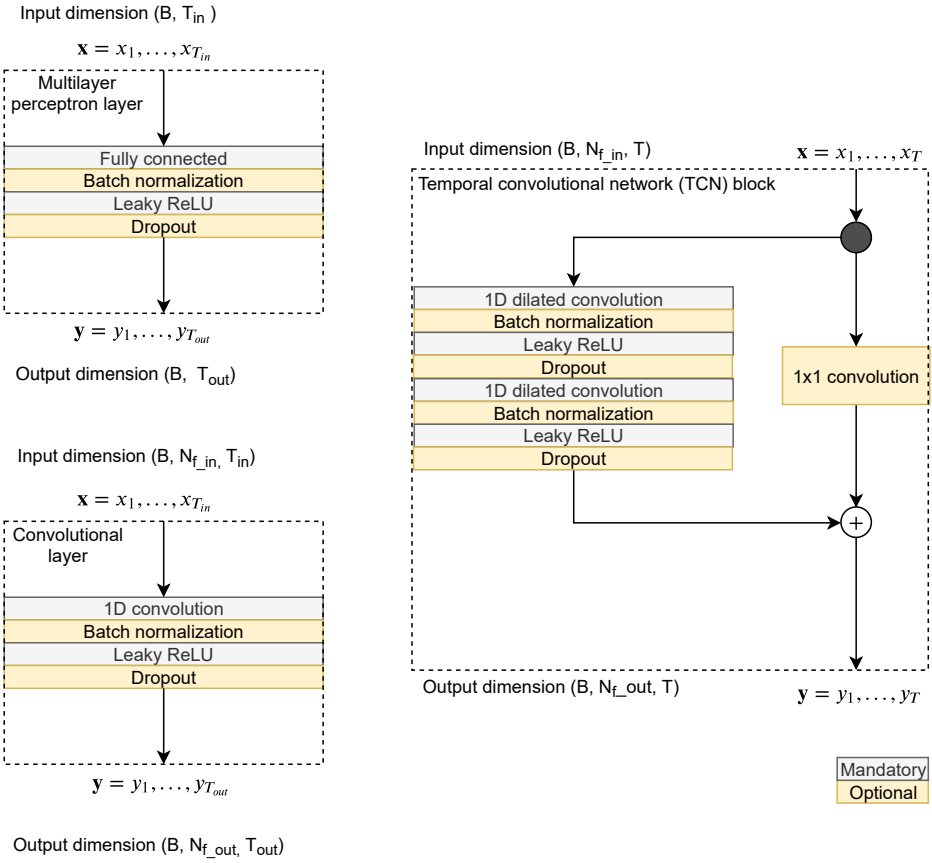


Figure 6.2: *Various types of hidden network layers utilized for construction of network architectures. Top left: Multilayer perceptron (MLP) layer. Bottom left: Convolutional layer. Right: Temporal convolutional network (TCN) block. Figure adapted from [110].*

The causality of the convolutions assures that during convolutional computations output values $\mathbf{y}_{t_2}$ are only convolved with input values $\mathbf{x}_{t_1}$ where $t_2 \geq t_1$, effectively assuring that there is no leaking of future information into the past. Simple causal convolutions allow only for considering a history linear in network depth [24]. Stacking dilated convolutions [255] allows to effectively create a large receptive field with exponential dependency on the network depth. Dilated convolutions introduce a fixed step between two adjacent filter taps, which is defined by the dilation factor d (cf. Fig. 6.3a). Considering this dilation factor d, dilated convolutional operators $F(\cdot)$ can be defined as follows [24]:

$$F(s) = (x *_d f)(s) = \sum_{i=0}^{k-1} f(i)\, \mathbf{x}_{s-d \cdot i} \tag{6.13}$$

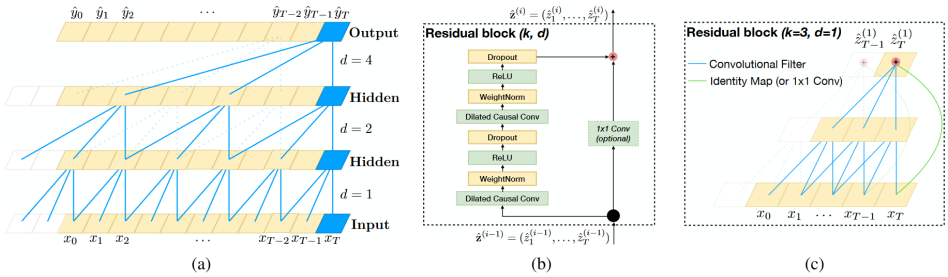


Figure 6.3: *Basic elements of TCN architectures as proposed in [24]. (a) Stacked dilated causal convolutions with increasing dilatation factors* $d = 1, 2, 4$ *and filter kernel size* $k = 3$*. (b) Residual block using two stacked dilated causal convolutional layers as depicted in Subfigure (a). The* 1×1 *convolution bypass allows directly adding inputs and outputs of the stacked dilated causal convolutional layers when exposing different dimensions. (c) Residual block with filter size* $k = 3$ *and dilation factor* $d = 1$*. Blue lines are related to convolutional filters, the green line illustrates the* 1×1 *convolution mentioned in Subfigure (b). Figure reproduced from [24] with kind permission of the authors.*

Here, a convolutional filter $f : \{0, \ldots, k-1\}$ of kernel size k is applied to an input vector $\mathbf{x}$. The equation illustrates the dependency of the filtering result at index s on input vector values at fixed step indices $s - d \cdot i$, with i denoting indices of the convolutional filter elements. The effective receptive field of this single convolutional filter computes to $(k-1)d$. Typically, multiple convolutional filters are stacked in a single TCN residual block in order to obtain a larger receptive field (cf. Fig. 6.3b with two stacked dilated convolutional filters). Successive to dilated convolutional filtering, batch normalization, Leaky ReLU activation and dropout are applied similar to MLP layers and convolutional layers.

Residual connections [95] add a second branch in the TCN residual block, creating a bypass to the series of transformations (i.e., dilated convolutional filtering, batch normalization, Leaky ReLU activation and dropout) applied to input tensors $\mathbf{X}$. Residual connections allow to apply transformations similar to the identity mapping to the input rather than the series of transformations mentioned before, which has recently shown to benefit the learning of very deep networks [95]. At the output of the TCN block, the outcomes of both branches (i.e., the series of transformations applied in the left branch and the unaltered input tensor $\mathbf{X}$) are summed element-wise. As unaltered input tensors $\mathbf{X} \in \mathbb{R}^{T \times N_f^{in}}$ and outcomes $\tilde{\mathbf{X}} \in \mathbb{R}^{T \times N_f^{out}}$ of the transformations in the left branch could have different channel dimensions, a 1×1 convolution (i.e., with a kernel size $k = 1$) needs to be added to the residual branch, which ensures that element-wise addition is applied to tensors of the same dimension. This 1×1 convolution has a number of output channels identical to the number of output channels N_f^{out} of the transformation series branch. Thus, when the output of the series of transformations

Table 6.2: List of compared neural anomaly detection models

Model	Loss function	Network architecture
AE model	AE loss (Eq. 6.1)	MLP encoder-decoder
VAE model	VAE loss (Eq. 6.3)	Conv./FCN/TCN encoder-decoder
SVDD model	Soft-boundary (Eq. 6.4) One-class (Eq. 6.5) Semi-sup. (Eq. 6.6) Weakly sup. (Eq. 6.12)	MLP/conv./FCN/TCN encoder

in the left branch has a different dimension than the inputs of the right residual branch, residual inputs are downsampled in order to align to the output of the left branch.

6.2.3 Training and Hyperparameter Optimization

Building on the hidden layer types introduced in the previous subsection, various encoder and decoder networks are defined. The encoder's main purpose is finding a descriptive internal representation of the input time series $\mathbf{x} \in \mathbb{R}^T$ given by an embedding vector $\mathbf{p} \in \mathbb{R}^P$, i.e., encoders function as the feature extractor part of the network. The main purpose of decoder networks is to reconstruct time series $\hat{\mathbf{x}}$ from the embedded representation vector $\mathbf{p}$ as closely as possible. Such decoder networks are necessary when training with either the AE or VAE loss function. For a detailed description of encoder networks and decoder networks, the interested reader is referred to Sections B.1 and B.2 in the appendix, respectively.

Encoders and decoders are combined into networks with an encoder-decoder architecture and pre-trained with the VAE loss function defined in Eq. 6.3. Afterwards, the encoders with weights initialized by this pre-training can be trained with various loss functions derived from the Deep SVDD loss proposed in [207]. Thus, pre-training allows finding a good initialization of the network weights, from which the estimate of the enclosing hypersphere center $\mathbf{c}$ is found by a forward pass through the encoder network. A descriptive summary of network architectures and loss functions considered in the upcoming experiments is listed in Table 6.2. MLP encoder-decoder networks serve as baseline among neural network architectures and are compared to multiple convolutional encoder-decoder networks. For this, various convolutional encoder networks (standard convolutional encoders, fully convolutional networks (FCNs) and TCNs) are combined with the convolutional decoder.

Encoder-decoder architectures trained with the VAE loss function necessitate a projection of the embedding vector $\mathbf{p}$ as output from the encoder network part to a latent space where the latent stochastic vector $\mathbf{z}$ is defined. Then, a similar

projection back to the embedding vector $\mathbf{p}$ which acts as input to the decoder network part has to be found. Both projections could be included into the encoder and decoder parts of the networks, but then the flexibility of combining encoders and decoders with the various loss functions as listed in Table 6.2 is lost. Thus, a separate projection network part is defined. For a detailed description of the projection network, the reader is referred to Section B.3 in the appendix.

Finally, despite the chosen types of encoders, decoders and loss functions, model performance is influenced by the applied training routine and hyperparameter optimization approach. Training as applied in this chapter closely resembles standard training routines minimizing loss functions via mini-batch SGD with a few adaptations. Hyperparameters are optimized either individually or jointly depending on the type of hyperparameters. For details both on training and hyperparameter optimization, the reader is again referred to Sections B.4 and B.5 in the appendix, respectively.

6.2.4 Label Generation via Probabilistic Graphical Models (PGMs)

So far, the presented neural models were considered unsupervised, i.e., not using any label information. Often, at least a small amount of labels is accessible. Deep semi-supervised models have shown to be remarkably good in leveraging even small amounts of label information. However, a good quality (i.e., reliability on correctness) of labels is often crucial. In this thesis, labels were so far obtained either via the live annotation approach or retrospective labeling by domain experts. Both labeling approaches were compared in Chapter 5. The evaluations in that chapter suggested retrospective annotations to be of higher quality than live annotations, thus better qualifying for inclusion in semi-supervised models.

In the former section, a weakly supervised extension of the semi-supervised Deep SVDD loss was proposed. This loss function incorporates information about label uncertainty when available by weighting labels regarding their probability of correctness $p(\tilde{y}_j)$. In order to generate labels in combination with such uncertainty estimates, methods based on the recent data programming paradigm [196] and probabilistic graphical models (PGMs) as label-generative models [21] can be applied. These methods allow for generating labels automatically and thus represent a third alternative labeling approach which comes without relying on direct involvement of human domain experts.

Data Programming

With the advent of deep learning techniques as powerful, learnable feature extractors, the main effort of preparing data for training of predictive models has shifted from hand-engineering features to labeling of data by human experts. As deep learning models require large sets of labeled data, this process can be quite expensive. Instead of human labeling of large data sets by domain experts, Ratner et al. introduced the data programming paradigm in [196]. They rely

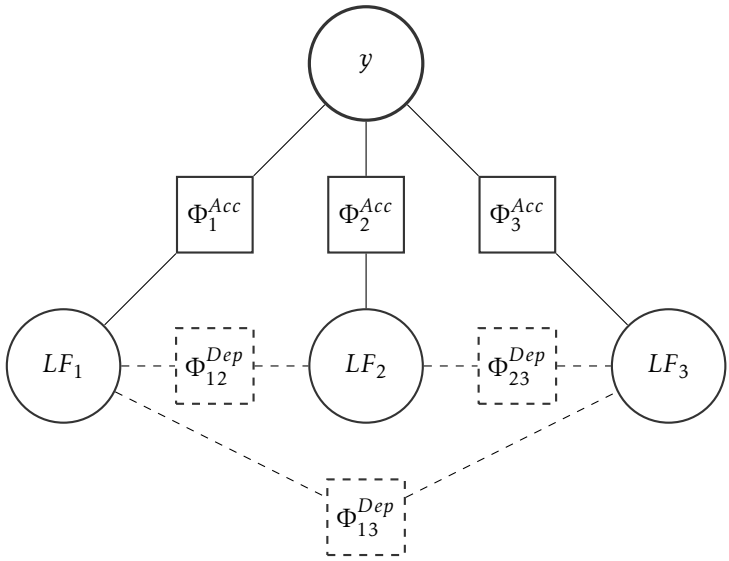


Figure 6.4: *Generic PGM modeling the structure of a label-generating process relying on multiple LFs. LFs have possible inter-dependency factors Φ^{Dep}. The true label y, which is never observed, is connected to the LFs via accuracy factors Φ^{Acc}. Figure inspired from [20].*

on incorporating domain expertise of users into definition of labeling functions (LFs).

These LFs allow labeling data sets in an automated and thus cheap manner but typically introduce a larger amount of noise into the labeling process than traditional domain expert labeling. In addition, LFs may conflict each other or may be highly correlated. For solving these issues, Ratner et al. propose to model the data set labeling process as a PGM, which allows modeling both accuracies and inter-dependencies of LFs as factors in a graph. Learning these PGM factors allows estimating final, more reliable labels by leveraging a multitude of noisy, potentially correlated or conflicting LFs.

Assumptions on the Structure of Label-Generating PGMs

A generic PGM which is general enough to model various assumptions about accuracies Φ^{Acc} of and dependencies Φ^{Dep} between label-generating LFs is illustrated in Fig. 6.4.

The overall goal of estimating optimal values for the accuracy and dependency parameters $\boldsymbol{\theta}$ is approached by minimizing the negative log marginal likelihood $-\log p_{\boldsymbol{\theta}}(\bar{\boldsymbol{\Lambda}})$ from a matrix of observed LF outputs $\bar{\boldsymbol{\Lambda}} \in \{-1, 0, 1\}^{m \times n}$ [21]:

$$\arg\min_{\theta} \; -\log \sum_{\mathbf{Y}} p_{\theta}(\bar{\boldsymbol{\Lambda}}, \mathbf{Y}) \tag{6.14}$$

Here, $\bar{\Lambda}_{i1}, \ldots, \bar{\Lambda}_{in}$ are obtained from n user-specified LFs $\lambda_1, \ldots, \lambda_n$. In general, LF outputs Λ_{ij} are in $\{-1, 0, 1\}$ for the binary classification case, corresponding to labels *false*, *abstaining* and *true*. Here, Λ_{ij} denote LF outputs in general, in comparison to observed matrix outputs $\bar{\Lambda}_{ij}$ Importantly, the true labels y_i are assumed to be never observed.

The joint probability $p_\theta(\mathbf{\Lambda}, \mathbf{Y})$ is modeled by the PGM being parameterized by $\boldsymbol{\theta} = \{\boldsymbol{\theta}^{Acc}, \boldsymbol{\theta}^{Dep}\}$, which quantify the accuracies of and dependencies between LFs, respectively. More formally, the PGM joint probability can be defined by [21]:

$$p_\theta(\mathbf{\Lambda}, \mathbf{Y}) \propto \exp\left(\sum_{i=1}^{m} \sum_{t \in T} \sum_{s \in S_t} \theta_s^t \phi_s^t(\Lambda_i, y_i) \right) \tag{6.15}$$

Here, T represents the set of considered dependency types. Dependencies $\phi_s^t \in \{0, 1\}$ exist both between true labels y_i and LF outputs Λ_i (i.e., being accuracies) as well as between LFs (correlations, conjunctions, etc.). S_t is a set of tuples of LF indices, indicating the participating LFs for each dependency of type $t \in T$. The factors $\theta_s^t \in [0, 1]$ quantify the degree of dependency t for the LF tuple s.

Optimizing the objective in Eq. 6.14 can be approached with standard SGD techniques but is computationally challenging for a generic PGM with possible interdependencies between LFs as defined in Eq. 6.15, as it involves Gibbs sampling for approximate estimation of the gradients of objective 6.14. When considering a sufficiently large number of LFs, this approach can become quickly computationally intractable, as the number of possible dependencies increases at least quadratically in the number of LFs [21].

A typical assumption to ease the computational complexity is conditional independence of the LF outputs $\mathbf{\Lambda}$ given the true labels Y [21]. Graphically, this results in omitting the dependency factors Dep between the LFs in Fig. 6.4. Formally, this results in accuracy dependencies $\phi_j^{Acc}(\Lambda_i, \lambda_i) \triangleq y_i \Lambda_{ij}$ between LF outputs Λ_{ij} and true labels y_i being the only dependencies T further considered. The joint probability then simplifies to

$$p_\theta(\mathbf{\Lambda}, \mathbf{Y}) \propto \exp\left(\sum_{i=1}^{m} \sum_{j=1}^{n} \theta_j^{Acc} \phi_j^{Acc}(\Lambda_i, y_i) \right) \tag{6.16}$$

Optimizing parameters $\boldsymbol{\theta}^{Acc}$ is then performed via the simplified negative log marginal likelihood

$$\arg\min_{\boldsymbol{\theta}^{Acc}} \; -\log \sum_{\mathbf{Y}} p_\theta(\bar{\mathbf{\Lambda}}, \mathbf{Y}), \tag{6.17}$$

which is equivalent to Eq. 6.14 but has to optimized only for accuracy parameters $\boldsymbol{\theta}^{Acc}$. The gradient of this marginal log likelihood objective 6.17 with respect to parameters θ_j^{Acc} can be formulated in closed form under the assumption of conditional independence of LF outputs [21].

Assuming conditional independence of the LF outputs given the true label in such a way is common and allows convenient estimation of parameters $\boldsymbol{\theta}$ without

necessitating to approximate gradients via Gibbs sampling. However, the conditional independence assumption is strong and often unjustified, e.g., LFs often incorporate similar information about the true label and are thus correlated. Then, neglecting these dependencies results in overconfident estimates of accuracy parameters θ_j^{Acc} for accuracy dependencies ϕ_j^{Acc} of these correlated LFs.

Instead of minimizing the negative log marginal likelihood, Bach et al. propose to instead minimize the negative log marginal pseudolikelihood of the outputs of single LFs λ_j [21]. Restricting the minimization on a single LF λ_j can be done by conditioning on the outputs of all other LFs $\lambda_{\setminus j}$. Here, the $\setminus$ operator denotes the absolute complement of j. In order to induce sparsity in the estimated parameters $\boldsymbol{\theta}$, Bach et al. add an ℓ_1 regularizer to the optimization problem, which can finally be stated as follows [21]:

$$\arg\min_{\boldsymbol{\theta}} \; -\log p_{\boldsymbol{\theta}}(\bar{\boldsymbol{\Lambda}}_j | \bar{\boldsymbol{\Lambda}}_{\setminus j}) + \epsilon \|\boldsymbol{\theta}\|_1 \tag{6.18}$$

$$= \arg\min_{\boldsymbol{\theta}} \; -\sum_{i=1}^{m} \log \sum_{y_i} p_{\boldsymbol{\theta}}(\bar{\Lambda}_{ij}, y_i | \bar{\Lambda}_{i\setminus j}) + \epsilon \|\boldsymbol{\theta}\|_1 \tag{6.19}$$

Here, $\epsilon > 0$ is a hyperparameter which needs to be optimized. This pseudolikelihood approach involving conditioning on all LFs but one allows computing gradients in polynomial time with respect to the number of LFs, data points, and dependency factors ϕ_s^t [21].

Weakly Supervised Deep Anomaly Detection

Based on a combination of label-generating PGMs with the weakly supervised Deep SVDD loss function proposed in Eq. 6.12, a weakly supervised anomaly detection model is described in the following paragraphs. The model is graphically summarized in Fig. 6.5.

Assume an unlabeled training data set $\mathcal{D}_N$ consisting of N raw data records in total. Then, a subset $\mathcal{D}_S$ of the training data is fed to the label-generative PGM described in the previous section. Based on labeling functions $\lambda_1, \ldots, \lambda_n$, weak labels $\Lambda_{i1}, \ldots, \Lambda_{in}$ (i.e., single LF outputs) are estimated. The label-generating PGM fuses these weak labels into stronger, probabilistic labels $p(\tilde{y}_i) \in [0, 1]$. These probabilities can be interpreted as uncertainty estimates $p(\tilde{y}_i)$ for labels $\tilde{y}_i$. Tuples $(\tilde{y}_i, p(\tilde{y}_i))$ for data records $\tilde{\mathbf{x}}_i \in \mathcal{D}_S$ are then used to train a weakly supervised neural anomaly detection model based on minimization of the weakly supervised Deep SVDD loss function (Eq. 6.12), which combines the information of unlabeled training data $\mathcal{D}_N$ with weakly labeled training data $\mathcal{D}_S$.

In this thesis, LFs are designed by individually thresholding n handcrafted single feature functions. For defining a threshold, scores of single features for the data set $\mathcal{D}_S$ are first z-score normalized. Based on the absolute values of these normalized feature scores, thresholds for each single feature LF are defined via

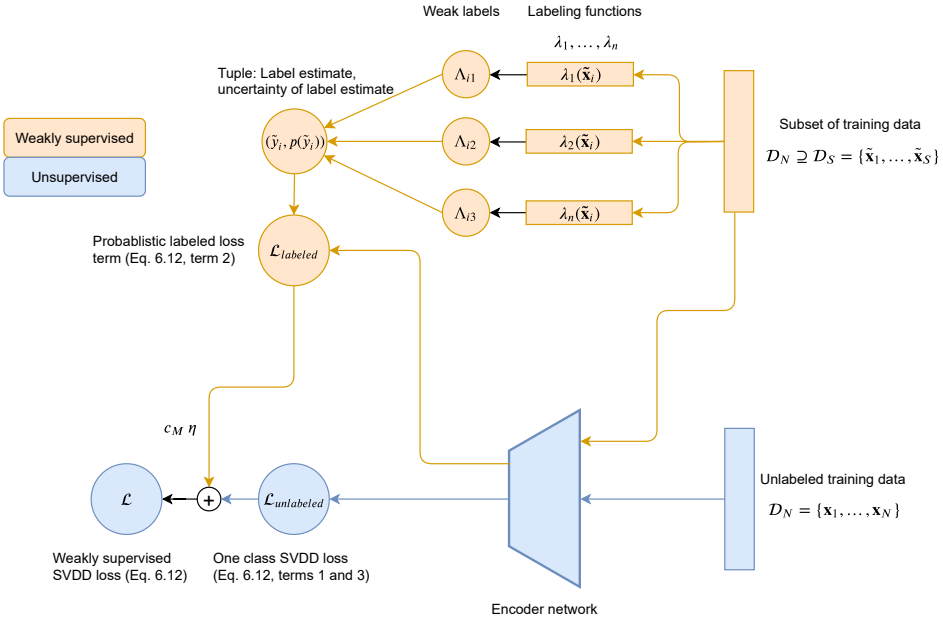


Figure 6.5: *Graphical summary of the proposed weakly supervised training procedure. Weak labels are created for a subset $\mathcal{D}_S$ of the unlabeled training data $\mathcal{D}_N$.*

Median Absolute Deviation (MAD) [140]:

$$\Lambda_{ij} = \begin{cases} +1, & \lambda_j(\tilde{\mathbf{x}}_i) \leq 1.5\ \text{MAD}(\lambda_j(\tilde{\mathbf{x}}_1), \ldots, \lambda_j(\tilde{\mathbf{x}}_S)) \\ 0, & 1.5\ \text{MAD}(\lambda_j(\tilde{\mathbf{x}}_1), \ldots, \lambda_j(\tilde{\mathbf{x}}_S)) < \lambda_j(\tilde{\mathbf{x}}_i) < 2.5\ \text{MAD}(\lambda_{\mathbf{j}}(\tilde{\mathbf{x}}_{\mathbf{1}}), \ldots, \lambda_{\mathbf{j}}(\tilde{\mathbf{x}}_{\mathbf{S}})) \\ -1, & \lambda_j(\tilde{\mathbf{x}}_i) \geq 2.5\ \text{MAD}(\lambda_{\mathbf{j}}(\tilde{\mathbf{x}}_{\mathbf{1}}), \ldots, \lambda_{\mathbf{j}}(\tilde{\mathbf{x}}_{\mathbf{S}})) \end{cases} \tag{6.20}$$

LF outputs $\Lambda_{ij} \in \{-1, 0, 1\}$ are interpreted as weak labels for the data records $\tilde{\mathbf{x}}_i$, where -1 denotes an anomaly and 1 normal data. $\tilde{\mathbf{x}}_i = 0$ means no label is assigned.

6.3 Results

In the following section, results for computational cost and performance of predictions with various neural anomaly detection models are summarized. First, the experimental setup is characterized, with a focus on describing the two evaluation data sets and a generic model size defined for fair comparison of all neural anomaly detection models applied. Afterwards, the performance of the various models on the two data sets is reported. Here, the performance of unsupervised models is compared to semi-supervised and weakly supervised model extensions.

Reported metric scores were originally presented in [110] supervised by the

author. Furthermore, except for Subsection 6.3.4, similar visualizations of learned embedding representations and predicted anomaly scores as illustrated in the upcoming experiments were presented in [110].

6.3.1 Experimental Setup

The data considered during experiments in the results section are represented by two data sets, a full data set considering all available data and a baseline data subset consisting of four successive days of data records from the full data set.

The full data set illustrates a large covariate shift of the normal data, i.e., the distribution of normal data during training and testing changes [242]. Learning a single anomaly detection which generalizes despite this covariate shift (i.e., without considering model adaptations in successive days) is challenging. Various encoder-decoder architectures trained with different loss functions as presented in Table 6.2 are considered in the upcoming experiments with the goal of finding an optimal encoder-loss function combination for tackling this challenge.

These encoder-loss function combinations are first benchmarked on the baseline data set, i.e., in a local subregion of the data with a reduced covariate shift. The goal of this first benchmarking on the baseline data set is comparing encoder-loss function combinations regarding their general capability to learn a powerful time series embedding representation of the data. Here, powerful representations are defined by both being able to cluster normal data close to each other and mapping abnormal data far away from these normal data.

The encoder-loss function combinations are then trained on the full data set. Training on the full data allows to judge which encoder-loss function combinations are optimal regarding handling the increased covariate shift in the full training data and generalizability of learned representations to previously unseen full test data.

Data Characteristics

The data considered in the experiments of this section consist of various days of vibration sensor data. The data were recorded with the vibration sensor at the "OP1" position attached to the same grinding machine as presented in Chapter 5. Data related to the warm-up class as presented in Chapter 5 were excluded from the experiments by application of simple energy threshold heuristics in order to avoid reporting overly optimistic performance results.

Characteristics of the data are listed in Table 6.3. The full data set consists of 47,484 data records in total recorded at 13 days. These 13 days were chosen due to a presence of machine damages or process anomalies visually confirmed by domain experts. Days of recording 12 and 13 were recorded during the user study conducted in Chapter 5. Expert-labeled days of recording 2 and 3 are the same ones as considered in Chapter 5 for benchmarking of anomaly detection algorithms.

Figure 6.6 illustrates the diversity of data for days of recording assembled in the full data set. The depicted curves represent the ensemble average of enve-

Table 6.3: *Data sets considered in this chapter. The full data set consists of 13 days of recording. The baseline data subset is marked in gray. Expert labeled days of recording are marked with ✓.*

Day of recording	Date of recording	Number of records	Expert labels
1	18-09-23	443	✗
2	18-09-24	2719	✓
3	18-09-25	2943	✓
4	18-09-26	5820	✗
5	18-10-11	5760	✗
6	18-10-12	4147	✗
7	18-10-29	5575	✗
8	19-01-29	4890	✗
9	19-01-31	3563	✗
10	19-02-13	1820	✗
11	19-02-14	4830	✗
12	19-04-15	2356	✗
13	19-04-16	2618	✗

lope signals extracted for these days of recording, the shaded areas mark the area of one standard deviation for these ensemble averages. The shaded areas thus illustrate the high intra-day variability of data records at several days of recording, while the change of ensemble averages across successive days of recording demonstrates the large covariate shift observed for the full data set. Being able to learn a compact representation of the normal data which is invariant to this covariate shift is the major challenge for the anomaly detection models evaluated in this section.

From this full data set, a baseline data (sub)set consisting of four successive days of recording is created. It is marked in gray in Table 6.3. This baseline data subset illustrates a reduced covariate shift compared to the full data as depicted in Fig. 6.7.

Both baseline data set and full data set are splitted into separate training, validation and test subsets. Training, validation and test data subsets are listed in Table 6.4. For the full data set, all available expert labeled data (days of recording 2 and 3) are used as test subset. 90% of the rest of the full data is considered for training, 10% for validation (hyperparameter optimization). For the baseline data set, 70% of the expert labeled data from days of recording 2 and 3 is used to create a test subset. Again, 90% of the rest of the baseline data is used for training, 10% for validation. Both training and hyperparameter optimization are performed in an unsupervised manner, thus expert labels of days of recording 2 and 3 are considered only for benchmarking of the presented model types on the test data.

As stated above, the fact that training, validation and test data for the baseline

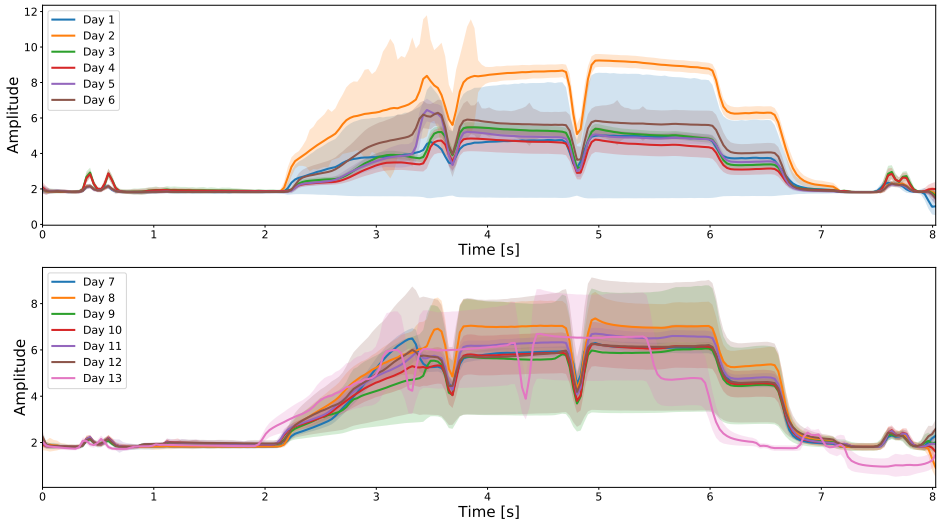


Figure 6.6: *Diversity of envelope signals extracted from records in the full data set. The curves represent ensemble averages of envelope signals of each day of recording, shaded areas display one standard deviation from these ensemble averages. A covariate shift across successive days of recording becomes apparent, most clearly for day 13.*

data set consist of successive days of recording leads to a reduced covariate shift and allows to evaluate the general capability of compared encoder-loss function combinations to learn a powerful embedding representations (i.e., independent from their capability to handle the large covariate shift in the full data).

Model Size

For the comparison of various combinations of encoder types and loss functions on both baseline and full data, a generic model size applicable to all types of encoder networks has to be defined. The model size is chosen to consist of three

Table 6.4: *Characteristics of training, validation and test data subsets of both full data (a) and baseline data (b)*

Data subset	Number of records
Training	37,628
Validation	4194
Testing	5662

(a) *Subsets of full data*

Data subset	Number of records
Training	7162
Validation	798
Testing	3965

(b) *Subsets of baseline data*

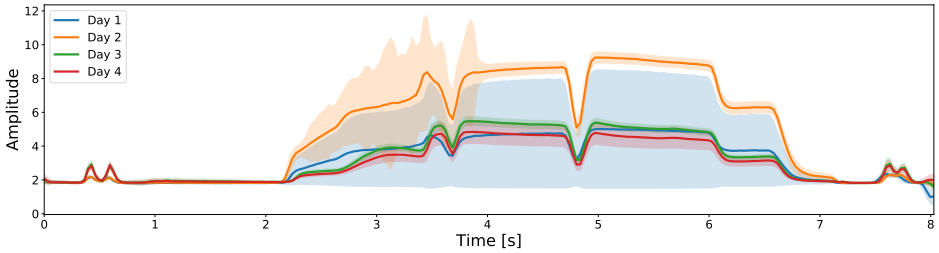


Figure 6.7: *Diversity of envelope signals extracted from records in the baseline data set. A reduced covariate shift of the data compared to Fig. 6.6 becomes apparent.*

Table 6.5: *Predictive cost of various encoder types for the defined compact model size (three hidden layers)*

Encoder	Training time [s]	Prediction time [s]	Number of parameters
Conv.	2.515	0.0429	37,648
TCN	12.52	0.1894	17,160
FCN	6.468	0.0711	17,936
MLP	0.234	0.00292	41,794

hidden layers only and an embedding dimension $p = 64$. This compact model size is chosen in order to address the challenges imposed by the concrete application: The risk of potential anomalies to cause severe machine damage necessitates short model execution times and response times. In addition, the anomaly detection model shall be deployed on the labeling prototype described in Chapter 5. The embedded nature of the labeling prototype thus imposes additional constraints on memory occupation. The compact model size values both computational and memory constraints.

The predictive cost of the considered compact models are listed in Table 6.5. Here, training time, prediction time per data record and number of network parameters that have to be stored are summarized. Training was performed on a CPU using an Intel Core i7-4810MQ processor with 32 GB RAM. Training times are reported per batches of 256 data records and as average training time per training epoch.

For convolutional, TCN and FCN encoders, the three hidden convolutional layers have kernel sizes $[32, 16, 16]$ and consist of 16 filters each. A compression factor $c_f = 2$ is used both for convolutional encoders and MLP encoders. When training with AE and VAE loss functions, symmetrical decoders are applied both for convolutional-type encoders and MLP encoders (i.e, expansion factor $e_f = 2$, same kernel sizes and amount of filters as for the encoders).

6.3.2 Anomaly Detection with Unsupervised Models

Learned representations and predictive results are first reported for various unsupervised neural anomaly detection models, whereas subsequent subsections cover semi-supervised and weakly supervised extensions. Models are compared quantitatively by reporting performance metric scores and qualitatively by visualizations of the learned embedding representations and predicted anomaly scores. Anomaly scores are computed via loss functions except for VAEs, where anomaly scores are represented by the reconstruction probabilities. In general, powerful embedding representations should illustrate low intra-cluster variance of normal data records and high distances of abnormal data to these normal data both in visualized embedding representations and anomaly scores.

PCA transformation is applied to the high-dimensional embedding spaces in order to allow for a meaningful visualization of the learned representations. In order to have a comparable visualization for all compared encoder-loss function combinations, the depicted PCA scores for the first and second principal components are min-max-scaled in order to map all PCA visualizations to a value range of $[0, 1]$. For anomaly scores, large outliers are removed via considering only the first 99 percentiles of data points. Then, min-max-scaling is applied as for the embedding space visualizations.

For quantitative comparison, F1 scores, precision, recall, average precision and receiver operating characteristic (ROC) area under curve (AUC) are reported. While average precision and ROC AUC are measures that summarize general performance of a model in an average value without necessitating to find an optimal choice of decision threshold on the computed anomaly scores, F1 scores, precision and recall necessitate choosing such a threshold. These thresholds are found for min-max-scaled anomaly scores via randomized stratified cross-validation with 5 folds and 100 runs on the test data sets of baseline and full data. F1 scores, precision and recall are then reported as average cross-validation scores. All performance measures are computed as class-weighted scores and by their scikit-learn implementations [184].

The various models are trained first on the baseline data set and on the full data set afterwards. In both cases, encoder types and unsupervised loss functions are varied in order to illustrate the differences in performance on the given data.

Training on Baseline Data Set

First, the embedding representations learned via unsupervised anomaly detection loss functions (AE, VAE, soft-boundary Deep SVDD and one-class Deep SVDD) are compared. This comparison is performed for compact models with FCN encoders. FCN encoders are chosen due to a combination of low memory / prediction time demands as listed in Table 6.5 and their excellent predictive performance as confirmed both by previous evaluations on a variety of time series applications in [266] and in the upcoming experiments.

Performance metrics for compact FCN models trained with various unsupervised loss functions are stated in Table 6.6. Here and in the following metric score tables, average precision is abbreviated by avg. prec. and best results are

Table 6.6: *Predictive performance of a compact FCN encoder model (three hidden layers) trained with various loss functions on the baseline data set*

Metric	Autoencoder	Variational Autoencoder	Soft-boundary Deep SVDD	One-class Deep SVDD
F1	0.4961	0.4961	0.9555	**1.0**
Precision	0.4922	0.4922	0.9987	**1.0**
Recall	0.5	0.5	0.9194	**1.0**
Avg. prec.	0.1354	0.4549	0.9985	**1.0**
ROC AUC	0.952	0.9161	1.0	**1.0**

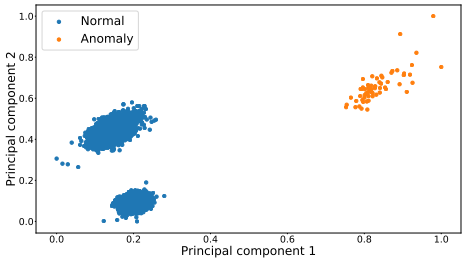
reported in bold font. Loss functions based on Deep SVDD clearly outperform AE and VAE loss functions, which are the most popular choices of anomaly detection loss functions. The one-class Deep SVDD loss function shows a slight performance advantage to the soft-boundary Deep SVDD loss function.

The learned embedding representations are visualized in Fig. 6.8. The results confirm the findings for performance metric scores: Deep SVDD loss functions (Fig.s 6.8a and 6.8b) being custom-built for anomaly detection succeed both in finding a compact description of the normal data and an embedding that maps anomalies far apart from these normal clusters. AE and VAE loss functions, which are more tailored for finding a compact representation of the complete data, map anomalous data close to normal clusters (AE, Fig. 6.8c) or seem trying to model them as part of the normal clusters (VAE, Fig. 6.8d).

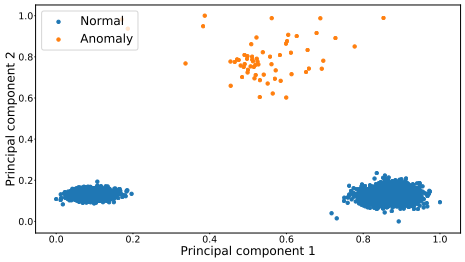
These differences in objectives of the loss functions are confirmed by the anomaly scores illustrated in Fig. 6.9. The good separation of anomalies (red patches) from normal data (gray patches) of Deep SVDD loss functions becomes apparent (Figures 6.9a and 6.9b). The one-class Deep SVDD loss function adapts better to the covariate shift in the normal data (smaller intra-cluster variance for anomaly scores of normal data, maximum anomaly scores for normal data below 0.2). On the other hand, while AE and VAE loss functions succeed in identifying anomalies by assigning high anomaly scores, normal data cannot be clearly separated from these abnormal data (i.e., a subset of normal data records are assigned high anomaly scores in the range of true anomalies).

In the following paragraphs, encoder types are varied in order to compare their predictive performance on the baseline data set. All of them were trained with the one-class Deep SVDD loss function, which proved most suited among the previously compared loss functions (cf. Figures 6.8, 6.9 and Table 6.6) and matches the characteristics of the data in this chapter well (the vast majority of data is normal, thus learning an optimal representation focusing on the normal data is sensible).

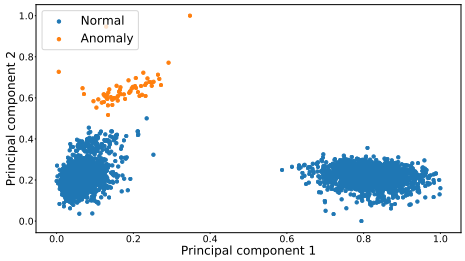
Training with the one-class Deep SVDD loss function resulted in perfect scores of 1.0 for all encoder types (convolutional, FCN, MLP and TCN) and performance metrics on the baseline data set. This result stresses the dominant influ-



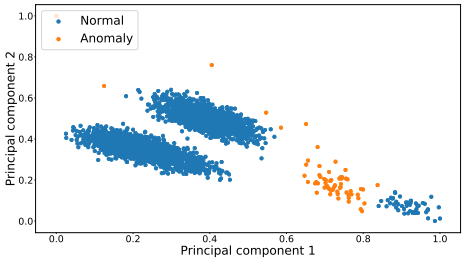
(a) One-class Deep SVDD loss



(b) Soft-boundary Deep SVDD loss

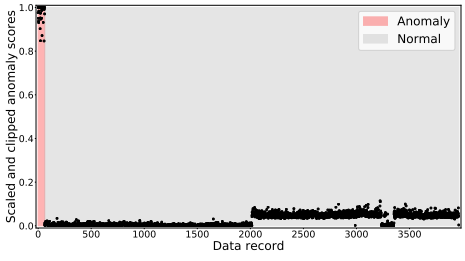


(c) Autoencoder (AE) loss

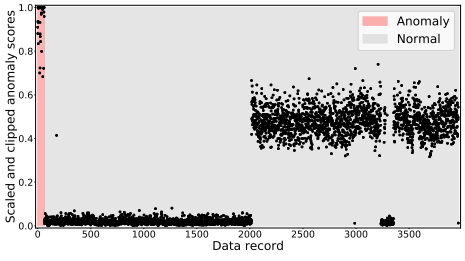


(d) Variational autoencoder (VAE) loss

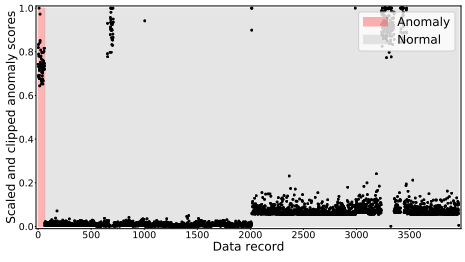
Figure 6.8: *Embedding spaces (FCN encoders, baseline data set)*



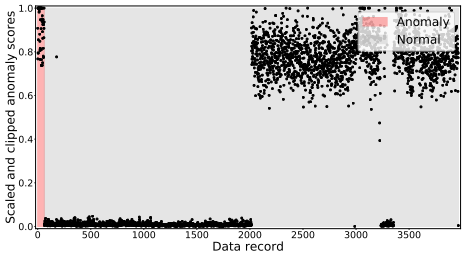
(a) One-class Deep SVDD loss



(b) Soft-boundary Deep SVDD loss

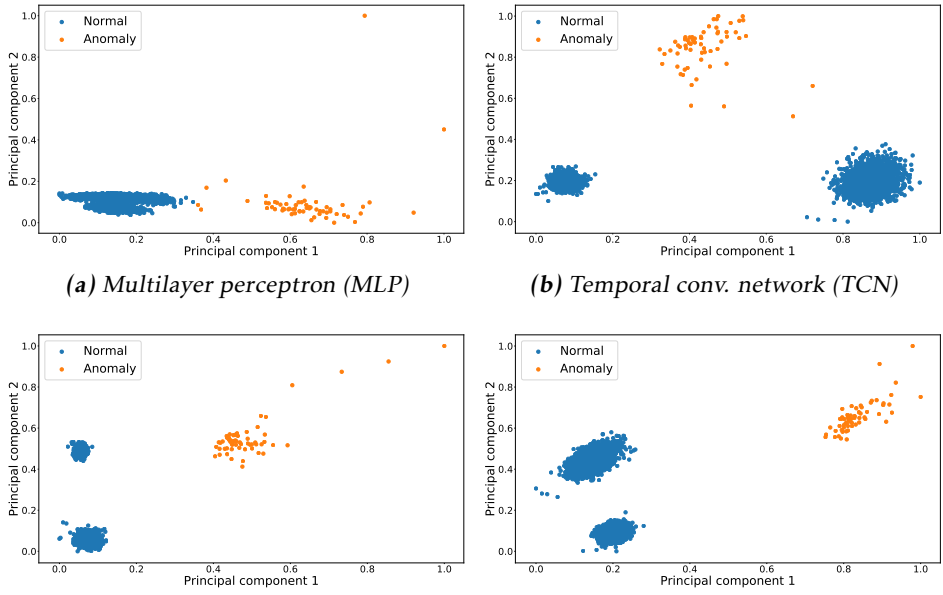


(c) Autoencoder (AE) loss



(d) Variational autoencoder (VAE) loss

Figure 6.9: *Anomaly scores (FCN encoders, baseline data set)*



(a) Multilayer perceptron (MLP)

(b) Temporal conv. network (TCN)

(c) Convolutional encoder

(d) Fully convolutional network (FCN)

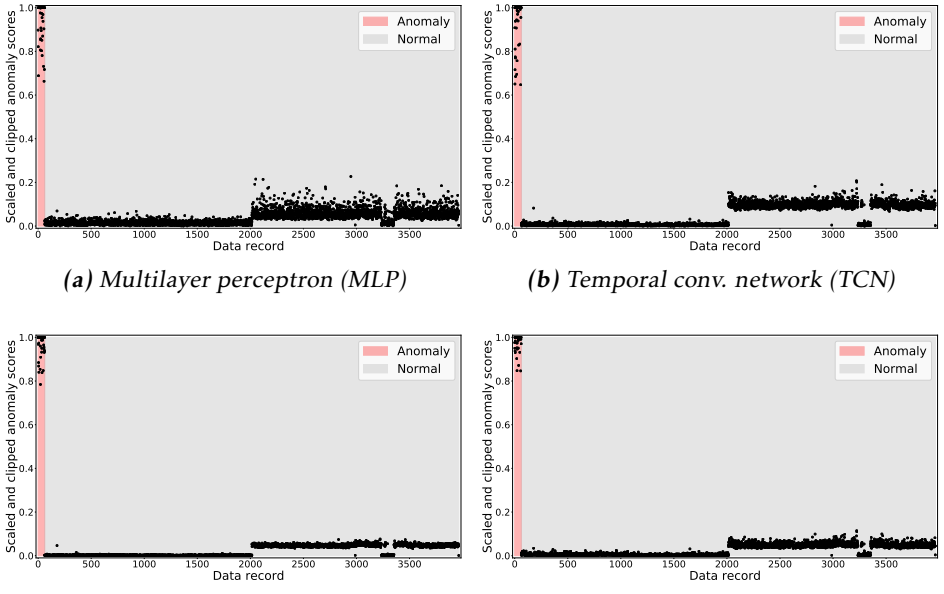
Figure 6.10: *Embedding spaces (One-class Deep SVDD loss, baseline data)*

ence of choice of loss function: When an appropriate loss function tailor-made to anomaly detection applications is chosen, design choices regarding encoder types seem to be of minor influence.

This finding is confirmed by the embedding space visualizations in Fig. 6.10 and the anomaly scores illustrated in Fig. 6.11. The learned embedding representations allow for a compact description of normal clusters while clearly separating anomalous data for all encoder types. Only for MLP encoders, a few anomalous records are mapped close to the normal clusters. The most compact normal clusters and clearest separation between normal and abnormal data are obtained with FCN encoders and convolutional encoders. This is confirmed by the anomaly scores depicted in Fig. 6.11, illustrating small normal intra-cluster variance and high distances between anomaly scores for normal and abnormal data.

Training on Full Data Set

The previous results suggested perfect performance of various encoder types even with a compact (three-layer) model when being trained with the one-class Deep SVDD loss function. However, the results reported might be overly optimistic, as both training and test data for the baseline data set consist of envelope signals from similar, successive days of recording. In the following experiments, encoder-loss function combinations are thus evaluated on the full data set, in or-



(a) Multilayer perceptron (MLP)

(b) Temporal conv. network (TCN)

(c) Convolutional encoder

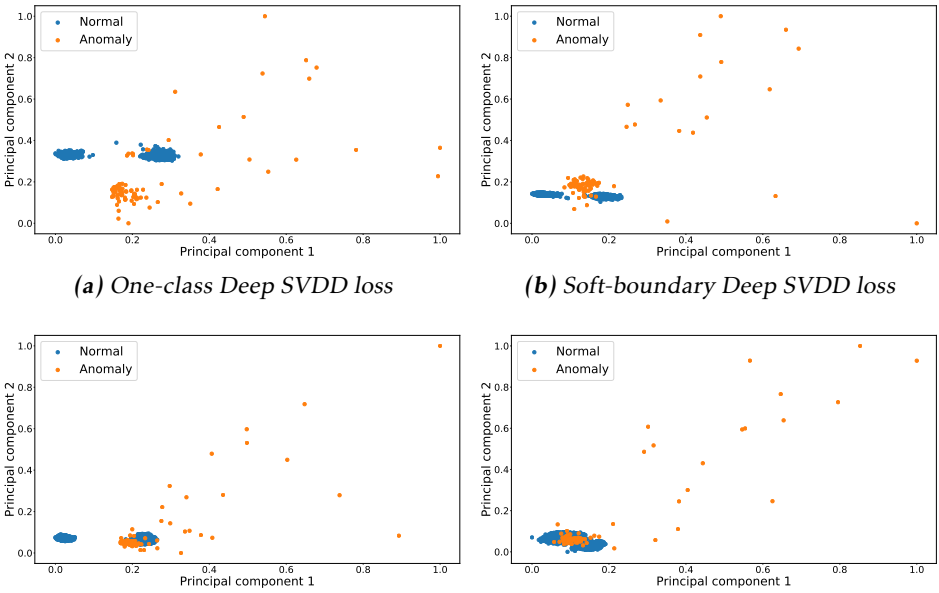
(d) Fully convolutional network (FCN)

Figure 6.11: *Anomaly scores (One-class Deep SVDD loss, baseline data set)*

der to obtain a more realistic evaluation of the capability of various encoder-loss function combinations to learn embedding representations that generalize across the increased covariate shift in the full data.

Similar as for the baseline data set, loss functions are varied for FCN encoders. Metric scores for prediction with these models are summarized in the upper part of Table 6.7 at the end of this subsection. The results show similar tendencies as for the baseline data set in Table 6.6: The one-class Deep SVDD loss outperforms other loss functions. This result is confirmed by visualizations of the learned embedding representations and predicted anomaly scores in Figures 6.12 and 6.13, respectively. AE and VAE encoders map both normal and most of the abnormal data to similar clusters in the embedding space being illustrated in Fig. 6.12. This finding is attributed to the objectives of AE and VAE loss functions to learn a compact representation of the data. On the other hand, embedding representations learned both by soft-boundary (SB) Deep SVDD loss function and one-class (OC) Deep SVDD loss function succeed in better separating normal from abnormal data. The separation with the latter loss function is again clearer than with the former as similarly reported for the baseline data. This finding is confirmed by the anomaly scores illustrated in Fig. 6.13, where only the one-class Deep SVDD loss function allows to clearly separate normal data from abnormal data. However, all loss functions depict a decreased separability performance compared to when being trained on the baseline data set (cf. Fig. 6.9). This is assumed to be due to the increased covariate shift in the full data. Accordingly, the embedding

representations being learned have to generalize across a much larger data set, thus illustrating more realistically the necessary capabilities of an anomaly detection model in order to perform well when being deployed and applied in-field.



(a) *One-class Deep SVDD loss*

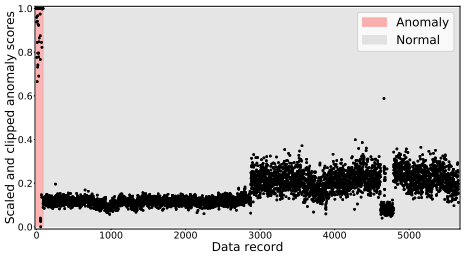
(b) *Soft-boundary Deep SVDD loss*

(c) *Autoencoder (AE) loss*

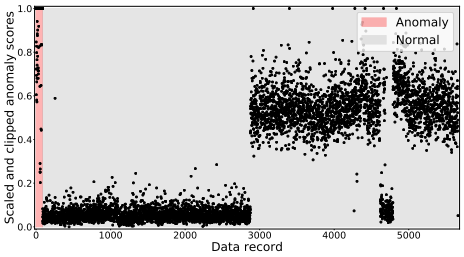
(d) *Variational autoencoder (VAE) loss*

Figure 6.12: *Embedding spaces (FCN encoders, full data set)*

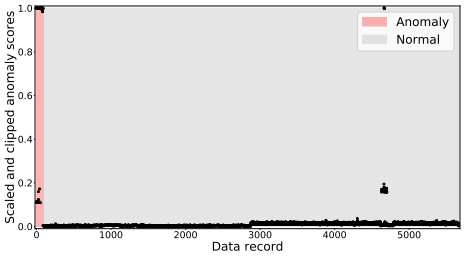
When varying the encoder types and training with the one-class Deep SVDD loss function on the full data set, metric scores as listed in the lower part of Table 6.7 are obtained. The results are more expressive than those for the baseline data, where all encoders reached perfect performance scores. For metric scores relying on a concrete choice of decision threshold (F1, precision and recall scores), FCN encoders and convolutional encoders outperform MLP encoders and TCN encoders. This is especially the case for F1 scores and recall scores, whereas precision scores are similar for all encoder types. Average precision and ROC AUC scores illustrate similar performance for all encoder types. The better performance of FCN encoders becomes understandable when considering the visualizations of learned embedding representations and predicted anomaly scores as illustrated in Figures 6.14 and 6.15. While MLP encoders and TCN encoders map normal data and most of the abnormal data to similar regions in the embedding spaces, convolutional encoders and especially FCN encoders succeed in separating normal data and the majority of abnormal data. This is confirmed by the anomaly scores depicted in Fig. 6.15, where convolutional encoders and FCN encoders allow for sensibly specifying a decision threshold on scaled anomaly scores for separation of normal and abnormal data while MLP encoders and TCN encoders do not.



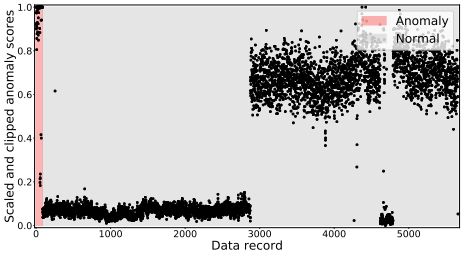
(a) One-class Deep SVDD loss



(b) Soft-boundary Deep SVDD loss

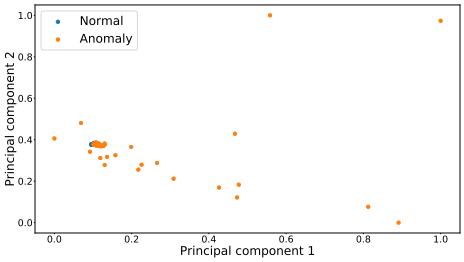


(c) Autoencoder (AE) loss

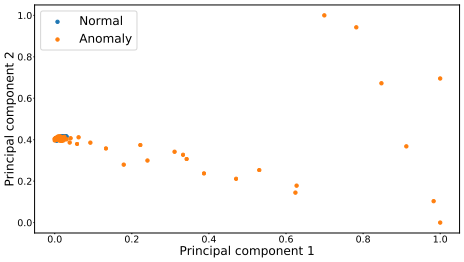


(d) Variational autoencoder (VAE) loss

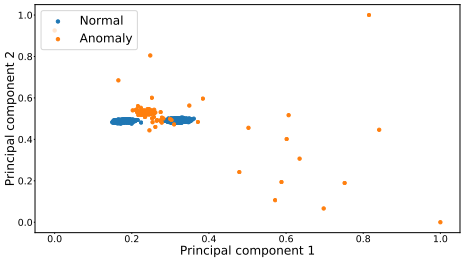
Figure 6.13: *Anomaly scores (FCN encoders, full data set)*



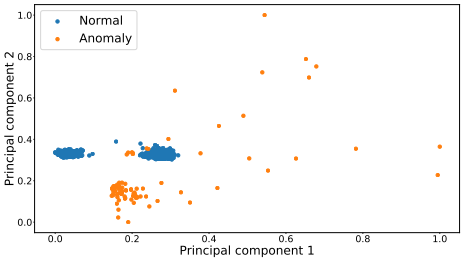
(a) Multilayer perceptron (MLP)



(b) Temporal conv. network (TCN)



(c) Convolutional encoder



(d) Fully convolutional network (FCN)

Figure 6.14: *Embedding spaces (One-class Deep SVDD loss, full data)*

Table 6.7: *Predictive performance on the full data set. Upper part of table: Compact FCN encoder model (three hidden layers) trained with various loss functions. Lower part of table: Various compact encoder models (three hidden layers) trained with the one-class (OC) Deep SVDD loss function.*

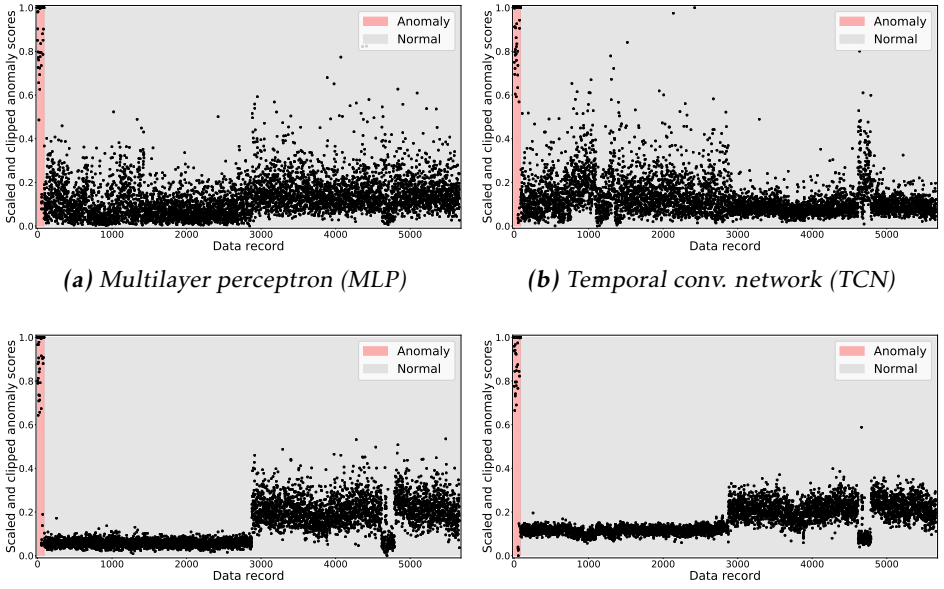
	F1 score	Precision	Recall	Avg. prec.	ROC AUC
Autoencoder	0.8733	0.9425	0.8232	0.8628	**0.9974**
Variational AE	0.4961	0.4922	0.5	0.8628	0.9609
SB Deep SVDD	0.4961	0.4922	0.5	0.7375	0.9516
OC Deep SVDD	**0.9662**	**0.999**	**0.9375**	**0.922**	0.9322
Conv. encoder	0.9595	0.9988	0.9261	**0.9226**	0.9492
FCN encoder	**0.9662**	**0.999**	**0.9375**	0.922	0.9322
MLP encoder	0.9120	0.9979	0.8636	0.9141	**0.958**
TCN encoder	0.9106	0.9824	0.8578	0.9028	0.9271

6.3.3 Utilizing Labels for Anomaly Detection Model Extensions

Neural models in general are known to highly benefit from including label information when accessible. Typically, a high quality of such labels is an important premise for an improvement of predictive performance of these (semi-)supervised model extensions. In this subsection, two different extensions of the unsupervised models evaluated in the former subsection are considered.

First, semi-supervised model extensions are presented. For semi-supervised extensions, 20% of the expert labels are used for training encoder models with the semi-supervised Deep SVDD loss function defined in Eq. 6.6. Afterwards, a weakly supervised model extension relying on automatically generated labels and trained with the weakly supervised Deep SVDD loss function proposed in Eq. 6.12 is evaluated. The training of this weakly supervised extension is performed as outlined in Subsection 6.2.4: For automatic generation of labels, LFs are created based on generic features which are extracted from the raw sensor data. The relevance of features is crucial for creating high-quality labels via these LFs. Here, the six most relevant features were selected from a set of approximately 600 generic features via SHaP [151], a recent technique for assigning each feature an importance value [110]. Then, a weak classifier is specified for each of these six features by learning a static threshold dependent on the feature's Median Absolute Deviation value (Subsection 6.2.4). These single-feature weak classifiers are then used as LFs.

Finally, a PGM as specified in [21] is learned to predict probabilistic label estimates $p(\tilde{y})$ by fusing the information from these LFs' outputs. Here, probability estimates $p(\tilde{y}_j)$ can be specified for both label classes +1 (normal) and −1 (abnormal). The class-conditional probabilities $p(\tilde{y}_j|\tilde{y}_j = +1)$ and $p(\tilde{y}_j|\tilde{y}_j = -1)$ are connected via $p(\tilde{y}_j|\tilde{y}_j = +1) = 1 - p(\tilde{y}_j|\tilde{y}_j = -1)$. Here, a probability $p(\tilde{y}_j|\tilde{y}_j = -1) = 1$ means the label-generating PGM is certain about the label estimate $\tilde{y}_j$ being an



(a) Multilayer perceptron (MLP)

(b) Temporal conv. network (TCN)

(c) Convolutional encoder

(d) Fully convolutional network (FCN)

Figure 6.15: *Anomaly scores (One-class Deep SVDD loss, full data)*

anomaly, for probabilities $p(\tilde{y}_j|\tilde{y}_j = -1) = 0$ the PGM is maximum uncertain about the label estimate $\tilde{y}_j$ being an anomaly.

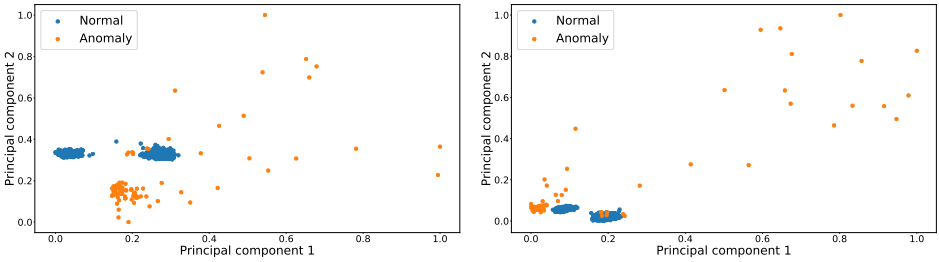
In the upcoming experiments, both predicting directly with the PGM by thresholding label probabilities (i.e., $p(\tilde{y}_j|\tilde{y}_j = -1) > 0.5$ predicted as an anomaly) and additionally incorporating the estimated label uncertainties inherent to the probabilities $p(\tilde{y}_j|\tilde{y}_j = +1)$ and $p(\tilde{y}_j|\tilde{y}_j = -1)$ into the process of learning a neural anomaly detection model with the weakly supervised Deep SVDD loss function are compared.

Semi-Supervised Model

A qualitative comparison of unsupervised and semi-supervised versions of compact FCN encoders is given by the learned embedding representations and predicted anomaly scores illustrated in Figures 6.16 and 6.17, respectively. For a quantitative comparison of both model versions, the reader is referred to Table 6.8 at the end of Subsection 6.3.3. There, a comprehensive comparison of metric scores for the unsupervised, semi-supervised and weakly supervised versions of the compact FCN encoders is presented.

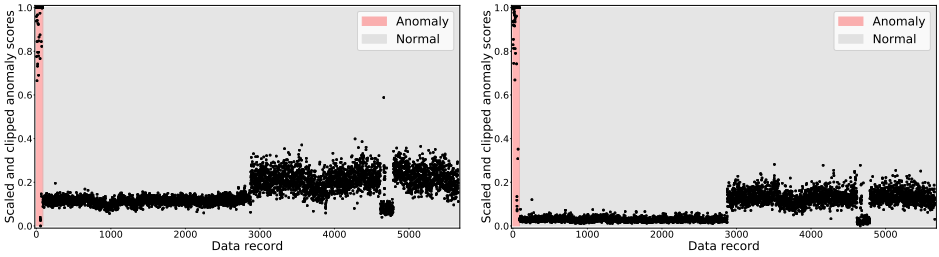
The embedding spaces of both unsupervised and semi-supervised compact FCN encoders in Fig. 6.16 are difficult to compare. Both models map the normal data to compact clusters while the majority of abnormal data is separated from both of these normal clusters. Considering the anomaly scores in Fig. 6.17 allows

for a better benchmarking of both model versions. As observable, using label information for the semi-supervised extension has two main effects. Firstly, the intra-class variance of anomaly scores for normal data is reduced compared to unsupervised models. Secondly, anomaly scores for normal data are predicted closer to zero for the semi-supervised version. This latter finding can be interpreted as an increased confidence of the semi-supervised model in its own predictions.



(a) One-class Deep SVDD loss *(b)* Semi-supervised Deep SVDD loss

Figure 6.16: *Embedding spaces (FCN encoders, full data set)*



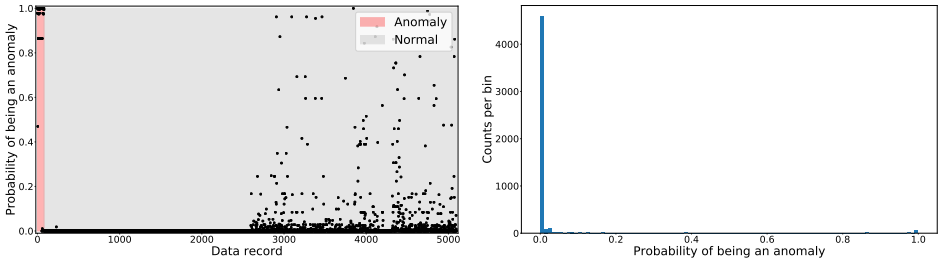
(a) One-class Deep SVDD loss *(b)* Semi-supervised Deep SVDD loss

Figure 6.17: *Anomaly scores (FCN encoders, full data set)*

Weakly Supervised Model

When no expert labels are accessible, labels can automatically be generated as outlined in the beginning of this subsection: A PGM as described in [21] can be learned to estimate probabilistic labels $p(\tilde{y}_j)$. Such a label-generating PGM is learned based on the purely unlabeled training data from the full data set in this subsection. Estimated probabilistic labels $p(\tilde{y}_j|\tilde{y}_j = -1)$ for the full test data are visualized in Fig. 6.18.

In Fig. 6.18a, the predicted label probabilities $p(\tilde{y}_j|\tilde{y}_j = -1)$ for the test data to be an anomaly are depicted. As described in the end of Subsection 6.2.4, the PGM



(a) Probabilistic labels *(b) Histogram of label probabilities*

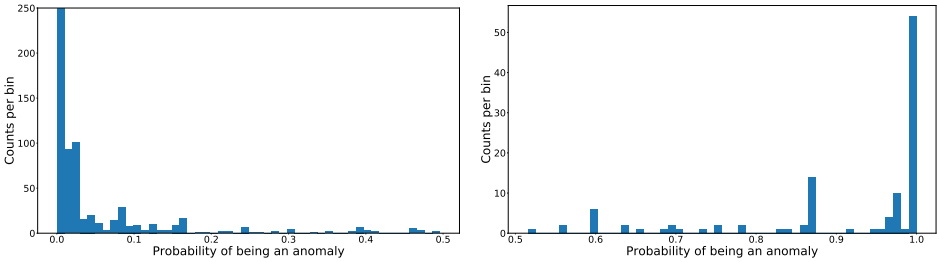
Figure 6.18: *Left: Probabilistic labels $p(\tilde{y}_j|\tilde{y}_j = -1)$ estimated by label-generating PGM. Right: Histogram of label probabilities.*

can refrain from assigning a label (i.e, $\tilde{y}_j = 0$). This is observable in the reduced number of test data records for which label probabilities are depicted. Sensible label probability estimates can be confirmed especially for anomalies (red patch) and the first day of normal test data (recording day 2 as listed in Table 6.3). This finding matches the anomaly scores illustrated for unsupervised models in previous experiments, where anomaly scores for recording day 3 depicted in the latter half of the anomaly score plots consistently illustrated higher anomaly scores and variance of the data.

Fig. 6.18b depicts a histogram of these probabilities $p(\tilde{y}_j|\tilde{y}_j = -1)$ illustrated in Fig. 6.18a. As observable, the vast majority of data is predicted with probabilities close to zero. Thus, the label-generating PGM is certain about the majority of test data records to be normal, which is sensible. Fig. 6.19 depicts the same histogram divided into two parts for better visibility. Figure 6.19a is truncated in vertical direction in order to better visualize less frequent probability estimates than in Fig. 6.18b. Label probabilities $p(\tilde{y}_j|\tilde{y}_j = -1) \leq 0.5$ as illustrated in Fig. 6.19a can be considered normal, probabilities $p(\tilde{y}_j|\tilde{y}_j = -1) > 0.5$ as depicted in Fig. 6.19b as abnormal. It can be observed, that most of the probability estimates concentrate at values of high certainty (i.e., 0 and 1). However, the label-generating PGM succeeds in estimating probabilities in between both values and can thus be interpreted as a well-calibrated probabilistic estimator [173].

The probabilistic labels as visualized in Fig. 6.18a can be used for prediction of the full test data in two ways. As a first option, the PGM itself can be interpreted as a classifier and the probabilistic labels in Fig. 6.18a as anomaly scores. When classifying an anomaly for scores $p(\tilde{y}_j|\tilde{y}_j = -1) > 0.5$, the metric scores listed in the upper part of Table 6.8 at the end of this subsection are obtained. The metric scores illustrate decent results but are inferior to the scores reported for the unsupervised neural anomaly detection models in previous subsections.

As a second option, a compact FCN encoder is trained with the weakly supervised Deep SVDD loss function proposed in Eq. 6.12. This necessitates tuples $(\tilde{y}_j, p(\tilde{y}_j))$ for training data records $\mathbf{x}_j$, where $p(\tilde{y}_j) \in [0, 1]$ is either of the probabilities $p(\tilde{y}_j|\tilde{y}_j = +1)$ or $p(\tilde{y}_j|\tilde{y}_j = -1)$ based on the class of labels $\tilde{y}_j \in \{+1, -1\}$. The

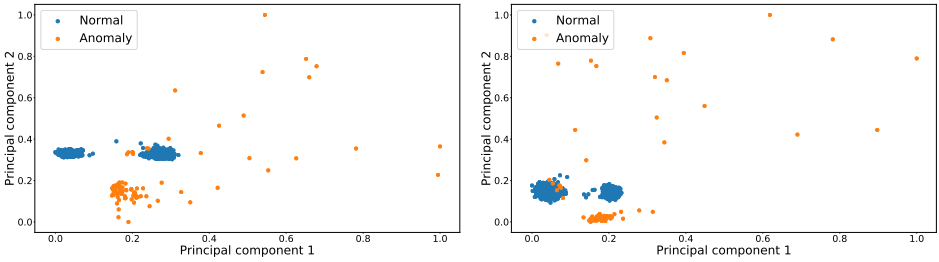


(a) Histogram of label probabilities ≤ 0.5 *(b) Histogram of label probabilities > 0.5*

Figure 6.19: *Histograms of label probabilities $p(\tilde{y}_j|\tilde{y}_j = -1)$. Left: Probabilistic labels considered normal ($p(\tilde{y}_j|\tilde{y}_j = -1) \leq 0.5$). Right: Probabilistic labels considered abnormal ($p(\tilde{y}_j|\tilde{y}_j = -1) > 0.5$).*

class label is again assigned by thresholding the probabilistic labels at 0.5 as specified before. Thus, neural anomaly detection encoders are provided with label uncertainty estimates $p(\tilde{y}_j)$ during training as opposed to when using the PGM as classifier (first option described above). A random subset of 10,000 $(\tilde{y}_j, p(\tilde{y}_j))$ tuples is used for training. Finally, predictions on the held-out test set of the full data can be computed with the trained weakly supervised FCN encoder.

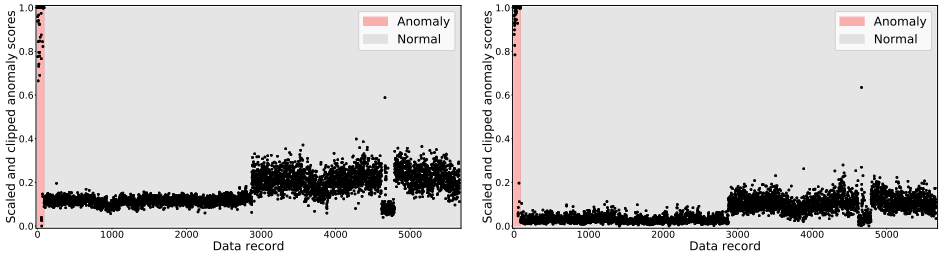
Training a compact FCN encoder with tuples $(\tilde{y}_j, p(\tilde{y}_j))$ as described results in learning an embedding representation as visualized in Fig. 6.20b. Predictions on the full test data with this weakly supervised FCN encoder results in anomaly scores as depicted in Fig. 6.21b. Both results are quite similar to the semi-supervised model extension: While the embedding spaces are difficult to compare, anomaly scores for the weakly supervised model illustrate a smaller variance in the anomaly scores for the normal data and higher confidence of the model (as confirmed by many near-zero anomaly scores for normal data and an increased distance between anomaly scores for normal and abnormal data).



(a) One-class Deep SVDD loss *(b) Weakly supervised Deep SVDD loss*

Figure 6.20: *Embedding spaces (FCN encoders, full data set)*

Finally, Table 6.8 confirms that both semi-supervised and weakly supervised



(a) One-class Deep SVDD loss *(b) Weakly supervised Deep SVDD loss*

Figure 6.21: *Anomaly scores (FCN encoders, full data set)*

Table 6.8: *Predictive performance of partly supervised models on the full data set. Upper part of table: Label-generating PGM used as a classifier (decision threshold at probabilities $p(\tilde{y}_j|\tilde{y}_j = -1) > 0.5$). Lower part of table: Compact FCN encoder models trained with unsupervised one-class (OC), semi-supervised (SSL) and weakly supervised (WSL) Deep SVDD loss functions.*

	F1 score	Precision	Recall	Avg. prec.	ROC AUC
PGM	0.8853	0.8327	0.9588	0.9082	0.9766
OC Deep SVDD	0.9662	0.999	0.9375	0.922	0.9322
SSL Deep SVDD	**0.979**	**0.9994**	**0.9602**	**0.9456**	**0.9784**
WSL Deep SVDD	**0.979**	**0.9994**	**0.9602**	0.9313	0.9716

extensions outperform the best-performing unsupervised FCN encoder (trained with the one-class Deep SVDD loss function) across all metric scores. Interestingly, training a model with automatically generated probabilistic labels and the weakly supervised Deep SVDD loss function results in similar metric scores as for the semi-supervised model trained with more than 1000 expert labels.

6.3.4 Anomaly Propositions with Neural Anomaly Detection Models

The previous experiments focused on comparing several combinations of encoder types and loss functions in order to find an optimal anomaly detection model. This comparison was approached both quantitatively (based on several performance metric scores) and qualitatively (based on visualizations of learned embedding spaces and predicted anomaly scores). Both computation of metric scores as well as visualizations of embedding spaces and anomaly scores were performed for the same test data as used for choice of an appropriate anomaly proposing

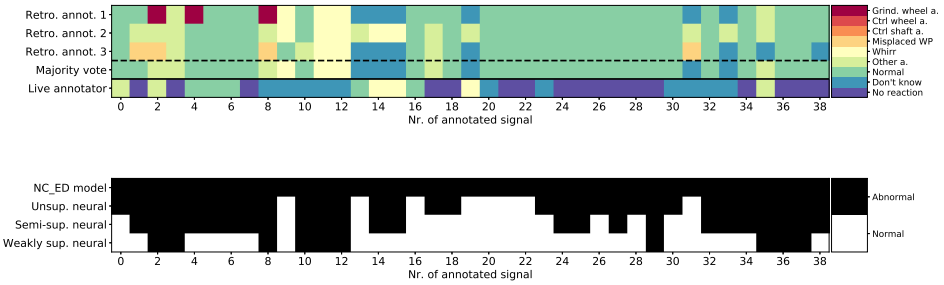


Figure 6.22: *Upper matrix: Agreement between multiple retrospective annotators and with live annotations of signals proposed as an anomaly in Chapter 5 (cf. Fig. 5.9). Lower matrix: Anomaly propositions by various anomaly detection models.*

model in the user study presented in the former chapter (cf. Subsection 5.5.1). However, Tables 5.1 and 5.2 confirmed excellent performance of various simpler models than the neural models presented in this chapter on these data. Then, during evaluation of the live annotations in Subsection 5.5.2, several experiments suggested these simple models not to be capable of reliably proposing actual abnormal data records, i.e., failed to model the complex normal behavior of the demo grinding machine due to covariate shift sufficiently well.

In order to find a more appropriate anomaly detection model, the various neural encoder-loss function combinations proposed in this chapter are compared with the simple NC_ED anomaly detector which was applied in Chapter 5. The NC_ED anomaly detection model is a nearest centroid (NC) model relying on Euclidean distance (ED) measures. The results of comparison with this model are reported for the live annotations collected in Chapter 5. For the evaluations, only the subset of live annotated data recorded with the "OP1" sensor position as described in Chapter 5 is considered, as the models in this chapter were trained and tested only for these OP1 sensor data.

Fig. 6.22 summarizes the agreement between human labels (retrospective and live annotated data, upper matrix) and anomaly propositions by four different models (lower matrix): The simple NC_ED anomaly detection model applied during the user study in Chapter 5 and the unsupervised, semi-supervised and weakly supervised neural anomaly detection models compared in the previous Subsection 6.3.3. Results are visualized in a similar form as in Fig. 5.9.

When comparing data records proposed as abnormal by these four models (black elements in lower matrix) with class labels assigned by the majority of retrospective annotators (fourth row of upper matrix) as feedback to these propositions, a higher agreement of neural models with annotators than that for the simple NC_ED models becomes apparent. The agreement increases when incorporating label information (lower matrix, third and fourth row), especially for the weakly supervised model. Quantitatively, when considering the majority vote of retrospective annotators as ground truth labels and comparing to anomaly

propositions, both precision scores and recall scores of neural anomaly detection models improve over scores of the simple NC anomaly detection model. Precision scores (as a measure for the false positive rate) improve from 24.24 % of the NC model to 66.67 % (unsupervised), 70.06 % (semi-supervised) and 83.72 % (weakly supervised) for the neural anomaly detection models. Similarly, recall scores (as a measure for the false negative rate) improve from 24.24 % of the NC model to 33.33 % (unsupervised), 48.48 % (semi-supervised) and 81.82 % (weakly supervised) for the neural anomaly detection models. The agreement with live annotations (upper matrix, fifth row) however stays low for all anomaly detection models. This substantiates both the power of neural anomaly detection models compared to the simple NC_ED anomaly detection model as well as the allegation of low reliability of live annotations collected as feedback to propositions by the simple NC_ED model applied in Chapter 5.

The agreement between neural anomaly propositions and the majority vote of retrospective annotators is especially high for the left part of the matrices, which consist of data recorded at April 15th and 16th, 2019 (cf. Fig. 5.8). These days of recording are characterized by visually confirmed machine damages as discussed in Chapter 5.

6.4 Conclusions

The major goal of this chapter was improving anomaly detection models over models from Chapter 5, using the ability of learning a time series representation that models the recorded normal data and thus normal state of the analyzed grinding machine as well as deviations from this normal state more reliably.

For this, multiple combinations of neural time series encoders and loss functions were compared. A compact (three-layer) model size was chosen, valuing constraints on memory space caused by the embedded nature of the demo evaluation system and the goal of short model execution times (due to the necessity of quick responses to potentially harmful anomalies).

The main challenge the anomaly detection models had to tackle is retaining an adequate representation of the normal data throughout the large observed covariate shift between days of recording in the evaluation data. But the compact model size chosen in this chapter would optionally allow for a retraining or adaptation of the anomaly detection models even on the embedded evaluation system.

The evaluation of various encoder-loss function combinations proved the superiority of neural anomaly detection models over simple anomaly detection models as applied in Chapter 5 on two data sets. Firstly, on a long-term (“full”) data set which illustrates the large covariate shift in the normal data occurring during in-field operation. The neural models learned on these full data succeeded in learning a representation generalizing the covariate shift as confirmed by visualized embedding spaces, anomaly scores and reported performance metrics (e.g., F1 scores of 96.6 % for best-performing unsupervised models and 97.9 % for best-performing semi-supervised and weakly supervised models). On the other hand, the simple anomaly detection models in Chapter 5 apparently failed

in generalizing this covariate drift as suggested by the high rejection rate of anomaly propositions. For a direct comparison, neural and simple anomaly detection models were compared on this second data set consisting of data records proposed as anomalous during the user study in the previous chapter. The latter data can be interpreted as the actual benchmark for appropriateness of anomaly detection models when being used as anomaly proposing model in the live annotation approach presented in Chapter 5: As evaluations of both retrospective and live annotations in the previous chapter revealed, both types of annotations illustrated a large disagreement with the propositions of potential anomalies by simple anomaly detection models. This chapter revealed that propositions by neural anomaly detection models align better with these retrospective annotations from the previous chapter: Precision scores improved from 24.24 % for the simple NC models applied in Chapter 5 to 83.72 % for the best-performing neural anomaly detection models evaluated in this chapter. In addition, recall scores improved from 24.24 % to 81.82 %.

The alignment increased when considering extensions of the unsupervised neural anomaly detection models that incorporate label information. Surprisingly, both semi-supervised extensions incorporating expert label annotations and weakly supervised extensions incorporating automatically generated labels showed a similar improvement. In addition, both semi-supervised models resulted in a similar improvement over unsupervised models when being compared on the full data set (F1 scores increased from 96.6 % to 97.9 % for both semi-supervised models). This similarity in predictive performance for semi-supervised and weakly supervised extensions occurred only when training the neural anomaly detection model extensions with the weakly supervised Deep SVDD loss function presented in this chapter.

Although neural anomaly detection models resulted in a higher agreement of anomaly propositions with retrospective label feedback, the alignment with live annotated data as collected in the previous chapter did not increase to a similar extent. Thus, the question remains, if live annotated data in general is unreliable or if the high disagreement was due to the simplicity and high FP rate of propositions. Alternative reasons encompass both a restricted visualization of the data considered during live annotation at the labeling prototype presented in Chapter 5 and restricted time during live annotation. All of these reasons can be circumvented by collecting potential abnormal data records proposed by the neural anomaly detection models in this chapter and presenting them for a retrospective annotation. This approach represents an alternative to live annotations with a similar labeling cost/quality ratio. Especially when using the weakly supervised neural anomaly detection model a near-optimal cost/quality ratio can be obtained: The weakly supervised neural anomaly detection model both improves the FP rate of the (lowest labeling cost) unsupervised neural models and the labeling cost of the semi-supervised neural model (relying on domain expert labels).

7

Summary

Present-day manufacturing processes in machine tool applications are exposed to high standards regarding efficiency and quality of workpieces' machining. Complying with these standards necessitates constant monitoring of the machining process and machine tools, as well as the machine parts they are assembled from. Such machine monitoring relies on evaluation of data recorded from sensors attached to key points of the machines.

Although sensor-based machine monitoring systems are already part of real-world production environments, they often involve a high degree of human intervention, which is both expensive and often error-prone. This lack of automation is despite several automated machine monitoring approaches found in literature and mainly due to three challenges:

- Machine monitoring systems striving for long-term deployment in real production environments need to cope with a **large covariate shift in sensor data** induced by changes of workpiece types and (frequent) process adjustments. This necessitates learning sensor data representations that generalize across this covariate shift, such that trained models still match the distribution of newly incoming test data.
- In addition, most performant predictive models across a wide range of time series applications proved to be (semi)-supervised, needing large sets of data annotated with trustworthy labels. Collecting such labels can be expensive due to occupation of machine operators and machine tools in experiments with non-standard process parameters, thus producing workpieces that can not be sold afterwards. In addition, machining with non-standard parameters can result in high risks of provoking severe machine damage. Both cost and risk of collecting anomaly labels often result in recorded **data sets equipped only with sparse and often noisy (i.e., not always reliable) labels**.

- Finally, most performant models often have high memory requirements, long training and model execution times or necessitate specific hardware for training. This conflicts with the **application demands for compact embedded sensor evaluation systems**, enabling retrospective equipment of existing machines with such systems. In addition, the risk of anomalies to cause machine damages induces **requirements of short model execution times**.

This thesis aimed at presenting automated, sensor-based machine monitoring systems addressing these challenges. Contributions made to that end are listed in the following section. All of them value memory constraints imposed by embedded evaluation systems and short model execution times required by the application.

7.1 Summary of Contributions

Chapter 3

Various segmentation approaches were evaluated on own sensor data recorded in real production environments. First, a novel segmentation model defined by a combination of GMMs and FSMs was introduced, mimicking HMMs but at reduced computational cost. Furthermore, an extension to the BOCPD [5] segmentation algorithm was presented. This extension aims at improved representing the cyclostationary behavior of the data-generating process in recurrent segments of the recorded sensor data. Modeling this cyclostationary behavior allowed for a more robust segmentation of data records and the successive extraction of features in the recurrent segments for an improved unsupervised detection of anomalies in the production process. In addition, a novel health indicator for tool condition monitoring represented by a custom-built feature relying on such recurrent segmentation was presented. Both the introduced segmentation methods and the novel tool condition health indicator claim to work independent from the choice of machine tool, process parameters and workpiece types, thus generalizing across the covariate shift observed in the recorded sensor data.

Chapter 4

Custom-built features were defined for condition monitoring of specific rotating (machine) parts. These features built on tracking of discrete frequency components related to rotating parts. Methods for a robust recovery and assignment of discrete frequency components to machine parts were presented. In addition, exemplary features for an unsupervised detection of imbalances in rotating machine parts and insufficient roundness of machined workpieces were introduced. As the presented features rely on domain expertise and physical understanding of the machine, they again claim to generalize across the covariate shift of evolving sensor data. The sensibility of these methods was again discussed for data recorded from a grinding machine in a real production environment.

Chapter 5

A novel approach for low-cost collection of large sets of expert labeled data in industrial scenarios by live and in situ annotation of sensor streams was presented. A longitudinal user study explored and evaluated the approach on a several-weeks-collection of sensor data recorded in a real production environment. Most importantly, the types of anomalies that can be labeled reliably with this approach were identified and influential factors on annotation reliability were discussed. A novel visualization and labeling prototype custom-made for annotation of sensor data in harsh industrial environments was presented, complemented by insights from the design process of this prototype gathered by exchange with domain experts (i.e., the industrial end users). This prototype deliberately deviates from the practice in frequent studies on annotation in medical and social applications, where labels are typically collected via a smartphone-based human–machine interface.

The prototypical system was developed and verified in a series of interviews with the industrial machine experts: Live visualization of the recorded sensor data at the prototype combined with timely alerts for potential anomalies (especially the severe classes whirring of workpieces and grinding wheel damage) allowed for saving a five-figure € amount per year in which the prototype was deployed and used. These cost savings compute solely from the absence of machine part changes, not having considered additional personal costs and expensive machine downtimes. The practical benefit the prototype proved in the production shop motivated machine operators to participate in live annotation of reported anomalies.

Chapter 6

As the user study from Chapter 5 suggested the applied anomaly detection models to be overly simple, a wide range of unsupervised neural anomaly detection models was evaluated. The models were defined from several combinations of time series encoder networks and anomaly detection loss functions, aiming at learning a time series embedding representation invariant to covariate shift in sensor data. All models were evaluated on a large corpus of sensor data recorded in the real production environment considered for the evaluations in Chapter 5, showing a large covariate shift. In addition, semi-supervised extensions of the models were evaluated. These extensions were both trained with expert labels and automatically generated weak labels. A novel weakly supervised anomaly detection loss function being able to estimate the increased uncertainty of automatically generated labels was used for training the latter of the semi-supervised extensions with the automatically generated labels. Surprisingly, this weakly supervised anomaly detection model proved competitive to the model trained with expert labels both regarding handling the covariate shift and in making better anomaly propositions than the simpler models shown in Chapter 5.

7.2 Conclusions and Outlook

The custom-built features proposed in Chapters 3 and 4 of this thesis allowed for an unsupervised detection of several specific machine monitoring tasks.

The extension of the BOCPD algorithm proposed in Chapter 3 enabled a robust yet computationally convenient detection of recurrent segments. Features extracted based on knowledge of these recurrent segments proved adequate both for estimation of tool condition (consistent increase of monotonicity and trendability performance measures) and unsupervised detection of sudden anomalies in the machining process (F1 scores of 99.05 % and 97.86 % for two selected predictive tasks). In general, alternative segmentation approaches explicitly exploiting the doubly cyclostationary structure of the data-generating process can be designed: Both data records and segments in these data records depict a recurrent structure. This could elegantly be represented in segmentation models with a hierarchical structure, e.g., extending the BOCPD approach or using hidden (semi) Markov models which are popular in speech segmentation applications [100].

In Chapter 4, the applicability of custom-built features for unsupervised detection of imbalances in rotating machine parts was validated. However, the applied approach building on tracking of discrete frequency components (DFCs) revealed susceptibility to spurious DFCs. Furthermore, recovering and assigning the single correct DFC to the matching machine part from a multitude of possible DFC candidates proved challenging. The recovery of continuous DFC tracks was complicated by the fractionation of tracks due to high amounts of noise in the sensor data. In general, extensions to improve the continuity of tracking can be possible by using more elaborate tracking methods like Markov renewal processes (MRPs) [243]. MRPs allow sleep states (i.e., discontinuities) in DFC tracks, thus potentially decreasing the fractionation of tracks and resulting in a smaller amount of continuous tracks. However, the two above mentioned challenges during the assignment of DFCs to matching machine parts (susceptibility to spurious DFCs and selecting the single correct DFC among a multitude of possible candidates) persists. This ultimately renders the approach of designing features building on DFCs being popular in the MHM community questionable.

The user study presented in Chapter 5 revealed diverging results on applicability of the live annotation approach. Live annotations from experts succeeded in identifying several actually confirmed anomalous events from the sensor data depicted via the developed labeling prototype. These confirmed anomalies were mainly representatives of anomaly classes with clearly deviating and well-known signal patterns (whirring of workpieces, grinding wheel damages). Other, more subtle signal deviations were typically not confirmed as anomalies by live annotators, resulting in a high rejection rate of potential anomalies proposed by the rather simple models (nearest centroid model, energy threshold heuristics). Besides the reduced time for annotations, main reasons for this high mismatch between propositions of and feedback to potential anomalies are assumed in two fields: The proposed anomalies being of classes with non-internalized patterns of signal manifestation and a limited visualization of sensor data at the labeling prototype screen. The former can be addressed by anomaly detection models be-

ing able to identify clusters of recorded data. Identifying clusters would allow for prompting a currently proposed potential anomaly together with similar formerly proposed and confirmed anomalous data records, allowing to internalize new patterns of manifestation of anomaly types previously unknown to machine operators. The limited visualization might be addressed by techniques highlighting the signal subregions most contributing to the currently predicted class of a data record under review. This highlighting of most abnormal signal subregions endows annotators with additional explanatory meta-information about prediction of potential anomalies, allowing for an enhanced visualization of more subtly deviating anomaly types.

Both of these extensions of the live annotation approach can be tackled with the advanced neural anomaly detection models presented in Chapter 6. Especially fully convolutional network (FCN) encoders combined with unsupervised, semi-supervised and weakly supervised versions of the Deep SVDD loss function [207] succeeded in learning a time series embedding representation that both generalizes across the observed covariate shift in the sensor data and allows to identify clusters in the learned embedding space. In addition, FCN encoders allow to use class activation mapping (CAM) [291] for highlighting of signal subregions most contributing to the predicted class for the current data record under review. The results in Chapter 6 revealed a good predictive performance of several model variants on the large recorded ("full") data set: The best unsupervised model (FCN encoder trained with one-class Deep SVDD loss function) reached an F1 score of 96.6 %, semi-supervised and weakly supervised FCN encoders both achieved F1 scores of 97.9 %. The increased predictive performance of these models compared to the simple models in Chapter 5 resulted in more reliable anomaly propositions: Precision scores improved from 24.24 % for the simple NC models applied in Chapter 5 to 83.72 % for the best-performing neural anomaly detection models evaluated in Chapter 6. In addition, recall scores improved from 24.24 % to 81.82 %. In general, other semi-supervised loss functions can be evaluated in order to further improve the predictive performance. Contrastive loss functions [91] and triplet loss functions [222] are among most popular recent choices. However, neither of them is custom-built for anomaly detection applications, typically necessitating data with more balanced classes than in anomaly or outlier detection scenarios.

More interestingly, using live annotations as labels in training the weakly supervised models would be appealing. Live annotations come without uncertainty estimates however, which were shown to be an influencing factor for success of training with the weakly supervised Deep SVDD loss function presented in Chapter 6. Such uncertainty estimates might be obtained by including live annotations as a labeling function in the label-generating probabilistic graphical model (PGM) described in Chapter 6. Good labeling functions should fulfill the two criteria of a good accuracy and good coverage of the data – both are not met by live annotations. Consequently, including live annotations as labeling function would reduce accessible data that can be used for PGM training, as live annotations only cover a small fraction of the data with labels, while generic feature labeling functions allow to estimate weak labels for all data records. This re-

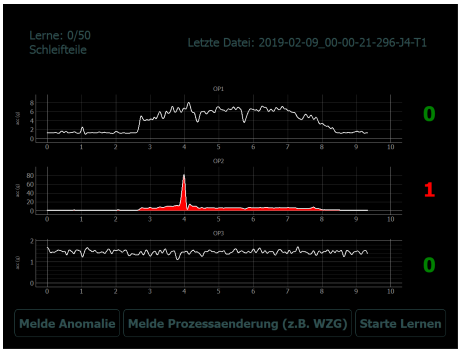
duced training data in turn would reduce the reliability of estimated accuracy factors and labeling function inter-dependency factors. In addition, a live annotation labeling function is likely to be of smaller accuracy than labeling functions based on appropriate features as confirmed by the high mismatch between live annotations and retrospective annotations illustrated in Chapters 5 and 6.

Appendix

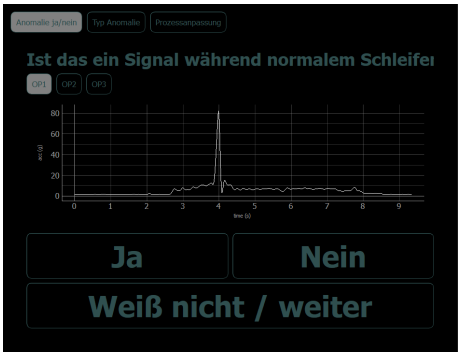
A

Appendix for User Study (Chapter 5)

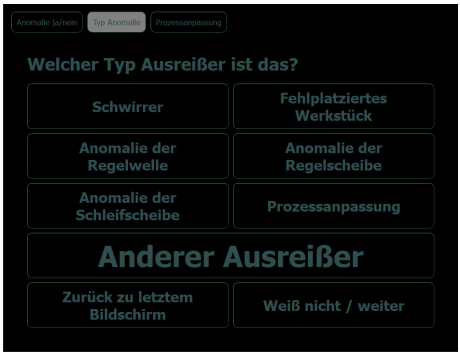
A.1 Original Version of Labeling Prototype Screens



(a) Default screen: Cont. visualization



(b) Screen 2: Anomaly (binary)



(c) Screen 3: Anomaly (multi-class)

(d) Screen 4: Process adapt. (multi-class)

Figure A.1: *Screens of the visualization and labeling prototype (original German version). A version of the screens translated to English can be found in Figure 5.3. A detailed description of the functional workflow of the screens can be found in Section 5.3.*

B

Appendix for Neural Anomaly Detection (Chapter 6)

B.1 Encoder Networks

In this subsection, various encoder architectures are described, building on the hidden layer types introduced in the previous subsection. The encoder's main purpose is finding a descriptive internal representation of the input time series $\mathbf{x} \in \mathbb{R}^{T \times N_c}$ given by an embedding vector $\mathbf{p} \in \mathbb{R}^p$, i.e., encoders function as the feature extractor part of the network. The encoder architectures in this section are adapted from the architectures mentioned in [266] (MLP, FCN) and [24] (TCN). In addition, a simple convolutional architecture is described, which uses similar 1D convolutional layer types as the FCN but progressively decreases the input sequence dimension T_{in}^l of each successive hidden layer l. Although the number of channels N_c for the input data considered in this chapter is one, the discussed architectures are general enough to be applied to arbitrary numbers of channels.

B.1.1 Multilayer Perceptron (MLP) Encoder

Fig. B.1 outlines the architecture of MLP encoders. Encoders are created by stacking L MLP layers. The dimension of input $\mathbf{x} \in \mathbb{R}^T$ to the first layer is progressively decreased by a fixed compression factor c_f, such that for layer l the output length computes to $T_{out}^l = \frac{T_{in}^l}{c_f}$.

B.1.2 Fully Convolutional Network (FCN) Encoder

The architecture of FCN encoders is illustrated in Fig. B.2. FCNs consist of L stacked convolutional layers, a global average pooling layer and a final 1×1 convolutional layer. FCNs were first used for time series applications in [266], where

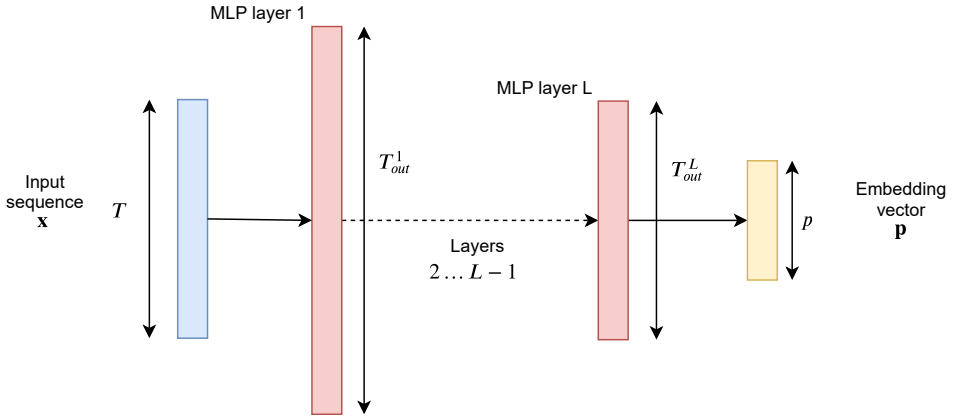


Figure B.1: *MLP encoder architecture. Figure adapted from [110].*

global averaging pooling was applied for the purpose of applying class activation mapping (CAM) [291], a technique allowing to identify subregions of time series contributing to the predicted class label. Different than for MLP encoders, the input sequence length T^1_{in} of the first convolutional layer is kept across all hidden layers. The number of filters N_f and the number of hidden layers L are set as recommended in [266]. The kernel size k is adapted in order to match the characteristics of the given data.

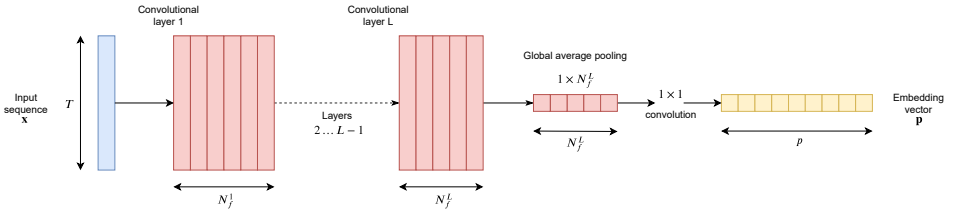


Figure B.2: *FCN encoder architecture. Figure adapted from [110].*

B.1.3 Convolutional Encoder

Similar to FCN encoders, convolutional encoders apply a series of L 1D convolutional layer transformations to the input data as illustrated in Fig. B.3. Other than for FCN encoders however, the output lengths T^l_{out} of layers l are progressively reduced by applying a constant compression factor c_f as for MLP encoders. The last of these convolutional layers produces an output of size $T^L_{out} \times N^L_f$, which is flattened and passed to a fully connected layer in order to produce an embedding vector $\mathbf{p} \in \mathbb{R}^p$.

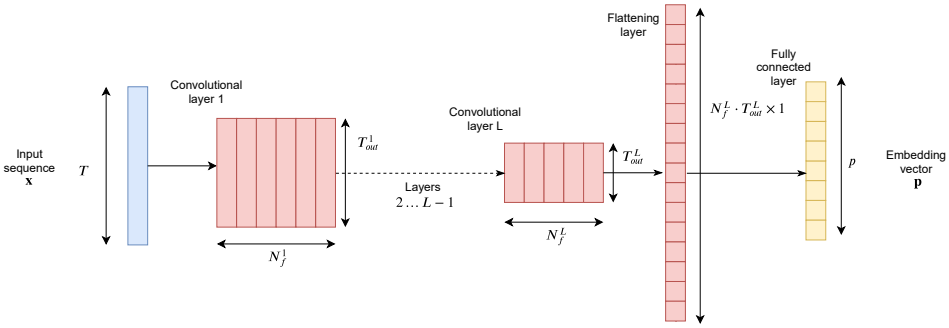


Figure B.3: *Convolutional encoder architecture. Figure adapted from [110].*

B.1.4 Temporal Convolutional Network (TCN) Encoder

Finally, the architecture of TCN encoders is depicted in Fig. B.4. Similar to FCN encoders, TCN encoders consist of L hidden blocks of similar structure, a successive global average pooling layer and a final 1×1 convolution. Also similar to FCN encoders, the output sequence length is kept constant across all hidden blocks. Other than for FCN encoders however, the hidden FCN blocks are replaced by TCN blocks, with the main difference of adding residual connections to each hidden block.

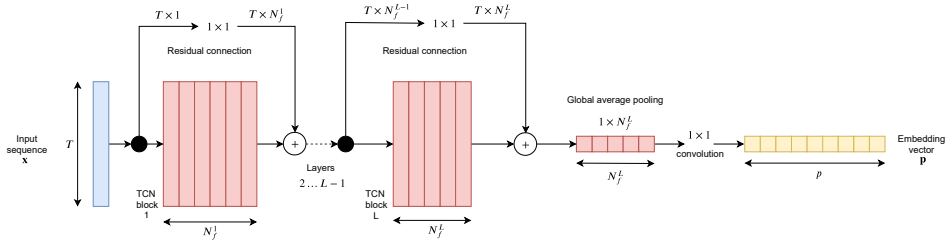


Figure B.4: *TCN encoder architecture. Figure adapted from [110].*

B.2 Decoder Networks

In this subsection, decoder network architectures considered in this chapter are described. Decoder networks are necessary for autoencoding structures, thus used only in combination with AE and VAE loss functions.

The main purpose of decoder networks is to reconstruct time series $\hat{\mathbf{x}}$ from the embedded representation vector $\mathbf{p}$ as closely as possible. In order to reconstruct time series $\hat{\mathbf{x}} \in \mathbb{R}^T$, successive hidden layers need to perform an upsampling of the lower-dimensional vector $\mathbf{p}$. This is obtained by an expansion factor $e_f = \frac{T^l_{out}}{T^l_{in}}$ that is kept constant for all layers l.

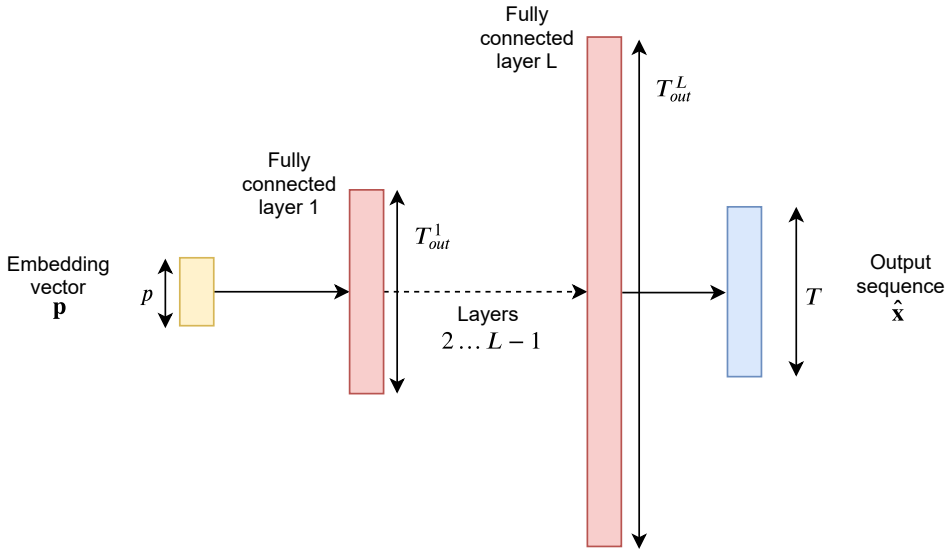


Figure B.5: *MLP decoder architecture. Figure adapted from [110].*

B.2.2 Convolutional Decoder

The considered convolutional decoder architecture is depicted in Fig. B.6. Upsampling is performed by transposed convolutional layers. Finally, a 1×1 convolution is applied to the output of the last transposed convolutional layer L in order to assimilate the number of channels of the reconstructed output $\hat{\mathbf{x}}$ to the number of channels $N_c = 1$ of the original input times series $\mathbf{x}$.

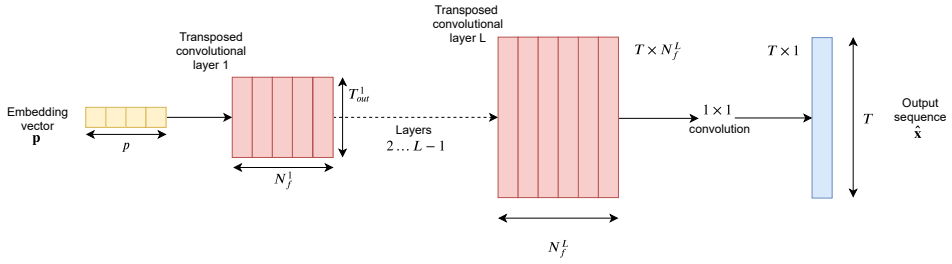


Figure B.6: *Convolutional decoder architecture. Figure adapted from [110].*

B.3 Variational Autoencoder (VAE) Projection Network

The projection network as illustrated in Fig. B.7 implements the reparameterization trick [124] by moving the sampling of $\mathbf{z}$ from hidden layers to an input layer. For the sampling of $\mathbf{z}$, two fully connected layers are implemented in order to represent the mean vector $\boldsymbol{\mu}$ and the standard deviation vector $\boldsymbol{\sigma}$ of the latent stochastic variable $\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$. Here, $\odot$ denotes an element-wise (Hadamard) product. Thus, $\mathbf{z}$ is assumed to be sampled from a multivariate normal distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2\mathbf{I})$ with diagonal covariance matrix [124], i.e., $\mathbf{I}$ denotes the identity matrix. The actual sampling from $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ is then represented by vectors $\boldsymbol{\epsilon}$ sampled from a standard normal distribution $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. This standard normal distribution is represented by another fully connected layer.

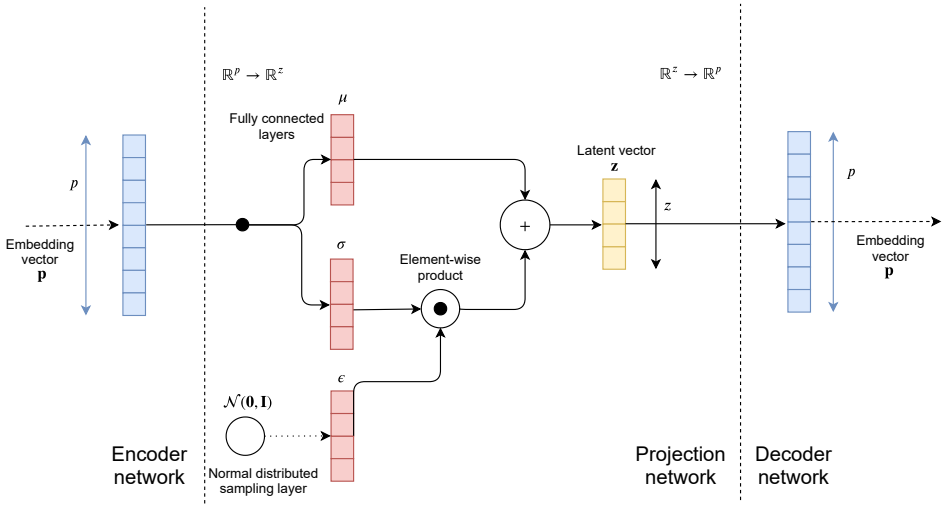


Figure B.7: *Projection network for the VAE encoder-decoder network. The separation of projecting the embedding vector* $\mathbf{p}$ *to the stochastic latent variable* $\mathbf{z}$ *and back from encoder and decoder parts of the network allows flexible combination of encoders and decoders with various loss functions. Figure adapted from [110].*

B.4 Training of Neural Anomaly Detection Models

Despite choice of loss function and network architecture, the applied training routine and the employed optimizer influence model performance. The neural models in this chapter are implemented in PyTorch [183]. Algorithm 3 summarizes the applied training routine. It resembles standard training routines minimizing loss functions via mini-batch SGD but illustrates a few adaptations.

Algorithm 3: Training Routine for Neural Anomaly Detection Models

Input: $\mathcal{D}_N^{train}$, $\mathcal{D}_M^{val}$
Output: $\mathcal{W}^*$

$\mathcal{W} \leftarrow \text{INIT}(\mathcal{W})$ ▷ Uniform Xavier weight initialization
for each epoch **do**
  for each mini-batch $\mathcal{B}$ in $\mathcal{D}_N^{train}$ **do**
    Compute training loss $\mathcal{L}_{\mathcal{B}}^{train}$ ▷ Forward pass
    Compute gradient $\Delta_{\mathcal{W}}\mathcal{L}_{\mathcal{B}}^{train}$ ▷ Backpropagation
    $\mathcal{W} \leftarrow \text{OPTIMIZER}(\mathcal{W}, \Delta_{\mathcal{W}}\mathcal{L}_{\mathcal{B}}^{train})$ ▷ Update weights
  end for
  for each mini-batch $\mathcal{B}$ in $\mathcal{D}_M^{val}$ **do**
    Compute validation loss $\mathcal{L}_{\mathcal{B}}^{val}$ ▷ Forward pass
  end for
  $\mathcal{L}_{\mu}^{val} \leftarrow \text{AVERAGE}(\mathcal{L}_{\mathcal{B}}^{val})$
  $\mathcal{L}_{\mu}^{train} \leftarrow \text{AVERAGE}(\mathcal{L}_{\mathcal{B}}^{train})$
  $\mathcal{L}_{\mu}^{val,*}, bStop \leftarrow \text{EARLYSTOPPING}(\mathcal{L}_{\mu}^{val})$ ▷ Update $\mathcal{L}_{\mu}^{val,*}$ and $bStop$
  if $bStop ==$ TRUE **then**
    Stop training
  end if
end for
return $\mathcal{W}^* \leftarrow \mathcal{W}$ for $\mathcal{L}_{\mu}^{val\,*}$

$\mathcal{D}_N^{train}$ is the training data set with N records in total and $\mathcal{D}_M^{val}$ is the validation data set with M records in total. The data consist of plain time series only, i.e., without providing any labels. Both training and validation data are split into a set of fixed-size mini-batches $\mathcal{B}$. For each epoch, training and validation losses of the mini-batches $\mathcal{B}$ are computed and averaged. The training ends either after having reached the prespecified number of epochs or when receiving a stop signal $bStop$. The value of this binary stop signal variable is specified by an early stopping routine, which updates the best average validation loss $\mathcal{L}_{\mu}^{val,*}$ observed during training and stops the training process when no further improvement (i.e., reduction of best average validation loss) has been obtained for a prespecified number of epochs. Early stopping thus acts as an additional regularizer to the weight regularizers applied with the individual loss functions, preventing from overfitting of the network parameters to the training data [190].

For each of the mini-batches $\mathcal{B}$ in $\mathcal{D}_N^{train}$, the training loss is computed by a forward pass through the network. Then, the gradients with respect to the weights $\Delta_{\mathcal{W}}$ are computed by backpropagation, using automatic differentiation capabilities of PyTorch [183]. Finally, the network weights are updated, using the Adam optimizer [122]. Network weights are uniformly Xavier initialized [85].

B.5 Optimization of Hyperparameters

The validation data $\mathcal{D}_M^{val}$ as denoted in Algorithm 3 are also used for hyperparameter optimization. Hyperparameters are optimized either individually or jointly depending on the type of hyperparameters.

Hyperparameters related to network architectural elements are the number of hidden layers L, the embedding dimension p, the latent dimension z and the kernel size k^l for convolutional filters of layer l. Except for the latter, these network architecture hyperparameters are individually optimized by variation of hyperparameter values and comparing to model performance. Kernel sizes k^l are optimized relying on best practices mentioned in [24, 266]. Then, network architecture hyperparameters with a tied dependency can be computed from these values of L, p, z and k^l. These dependent hyperparameters are the compression factor c_f and the expansion factor e_f as well as padding sizes pd^l and dilation factors d^l of layers l.

After optimization of hyperparameters related to the network architecture, hyperparameters related to loss functions and the training routine summarized in Algorithm 3 are optimized. These hyperparameters are mainly the learning rate η, size B of mini-batches $\mathcal{B}$ and the trade-off factor λ governing the influence of the weight regularizer $\sum_{l=1}^{L} \|\mathbf{W}_l\|_F^2$ utilized in most loss functions. These hyperparameters are tightly related [236], which is why they are jointly optimized by an algorithm leveraging a tree of parzen estimators [31]. For joint optimization of these latter hyperparameters, the Hyperopt [32] software package is used.

C

List of Figures

D

List of Tables

Bibliography

[1] Norm DIN 8580. Fertigungsverfahren - Begriffe, Einteilung, September 2003.

[2] J. V. Abellan-Nebot and F. R. Subirón. A review of machining monitoring systems based on artificial intelligence process models. *The International Journal of Advanced Manufacturing Technology*, 47(1):237–257, March 2010.

[3] R. Adams and B. Marlin. Learning time series detection models from temporally imprecise labels. In *Proc. of the 20th International Conference on Artificial Intelligence and Statistics*, AISTATS'17, 2017.

[4] R. Adams and B. Marlin. Learning time series segmentation models from temporally imprecise labels. In *Proc. of the 34th Conference on Uncertainty in Artificial Intelligence*, UAI'18, pages 1–10, 2018.

[5] R. P. Adams and D. J. C. MacKay. Bayesian online changepoint detection. Technical report, University of Cambridge, Cambridge UK, 2007.

[6] C. C. Aggarwal. Outlier ensembles: Position paper. *SIGKDD Explorations Newsletter*, 14(2):49–58, April 2013.

[7] C. C. Aggarwal. *Outlier Analysis*. Springer Publishing, Berlin Germany, 2nd edition, 2016.

[8] C. C. Aggarwal and S. Sathe. Theoretical foundations and algorithms for outlier ensembles. *SIGKDD Exploration Newsletters*, 17(1):24–47, September 2015.

[9] C. C. Aggarwal and P. S. Yu. An effective and efficient algorithm for high-dimensional outlier detection. *The VLDB Journal*, 14(2):211–221, April 2005.

[10] A. Agogino and K. Goebel. Milling data set. Website, 2007.

[11] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, December 1974.

[12] A. Akbari and R. Jafari. An autoencoder-based approach for recognizing null class in activities of daily living in-the-wild via wearable motion sensors. In *Proc. of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP'19, pages 3392–3396, 2019.

[13] E. Aksan and O. Hilliges. STCN: Stochastic temporal convolutional networks. *arXiv preprint arXiv:1902.06568*, 2019.

[14] F. J. Alonso and D. R. Salgado. Analysis of the structure of vibration signals for tool wear detection. *Mechanical Systems and Signal Processing*, 22(3): 735–748, April 2008.

[15] Y. Altintas and S. S. Park. Dynamic compensation of spindle-integrated force sensors. *CIRP Annals*, 53(1):305–308, 2004.

[16] S. Aminikhanghahi and D. J. Cook. A survey of methods for time series change point detection. *Knowledge and Information Systems*, 51(2):339–367, May 2017.

[17] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. OPTICS: Ordering points to identify the clustering structure. In *Proc. of the 1999 ACM SIGMOD International Conference on Management of Data*, SIGMOD'99, pages 49–60, 1999.

[18] I. Antoniadou, G. Manson, W. J. Staszewski, T. Barszcz, and K. Worden. A time–frequency analysis approach for condition monitoring of a wind turbine gearbox under varying load conditions. *Mechanical Systems and Signal Processing*, 64–65:188–216, December 2015.

[19] S. A. Aye and P. S. Heyns. An integrated gaussian process regression for prediction of remaining useful life of slow speed bearings based on acoustic emission. *Mechanical Systems and Signal Processing*, 84:485–498, February 2017.

[20] S. Bach, B. He, A. Ratner, and C. Ré. Structure learning: Are your sources only telling you what you want to hear?, 2017. URL `https://hazyresearch.github.io/snorkel/blog/structure_learning.html`.

[21] S. H. Bach, B. He, A. Ratner, and C. Ré. Learning the structure of generative models without labeled data. In *Proc. of the 34th International Conference on Machine Learning*, ICML'17, pages 273–282, 2017.

[22] M.-A. Badiu, T. Lundgaard Hansen, and B. Henri Fleury. Variational bayesian inference of line spectra. *IEEE Transactions on Signal Processing*, 65(9):2247–2261, May 2017.

[23] A. Bagnall and J. Lines. An Experimental Evaluation of Nearest Neighbour Time Series Classification. *arXiv preprint arXiv:1406.4757*, June 2014.

[24] S. Bai, J. Z. Kolter, and V. Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[25] W. U. Bajwa, J. Haupt, A. M. Sayeed, and R. Nowak. Compressed channel sensing: A new approach to estimating sparse multipath channels. *Proceedings of the IEEE*, 98(6):1058–1076, June 2010.

[26] A. M. Bassiuny and X. Li. Flute breakage detection during end milling using hilbert–huang transform and smoothed nonlinear energy operator. *International Journal of Machine Tools and Manufacture*, 47(6):1011–1020, May 2007.

[27] J. Bayer and C. Osendorfer. Learning stochastic recurrent networks. *arXiv preprint arXiv:1411.7610*, 2014.

[28] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, March 1994.

[29] T. Benkedjouh, K. Medjaher, N. Zerhouni, and S. Rechak. Health assessment and life prediction of cutting tools based on support vector regression. *Journal of Intelligent Manufacturing*, 26(2):213–223, April 2015.

[30] T. Benkedjouh, N. Zerhouni, and S. Rechak. Tool condition monitoring based on mel-frequency cepstral coefficients and support vector regression. In *Proc. of the 5th International Conference on Electrical Engineering*, pages 1–5, 2017.

[31] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems 24*, NIPS'11, pages 2546–2554, 2011.

[32] J. S. Bergstra, B. Komer, C. Eliasmith, D. Yamins, and D. D. Cox. Hyperopt: A python library for model selection and hyperparameter optimization. *Computational Science and Discovery*, 8(1):014008, January 2015.

[33] E. Berlin, K. Van Laerhoven, and B. Schiele. An empirical study of time series approximation algorithms for wearable accelerometers, November 2009.

[34] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, AAAIWS'94, 1994.

[35] B. Betea, P. Dobra, M.-C. Gherman, and L. Tomesc. Comparison between envelope detection methods for bearing defects diagnose. *IFAC Proceedings Volumes*, 46(6):137–142, May 2013.

[36] M. Betser. Sinusoidal polynomial parameter estimation using the distribution derivative. *IEEE Transactions on Signal Processing*, 57(12):4633–4645, December 2009.

[37] B. N. Bhaskar, G. Tang, and B. Recht. Atomic norm denoising with applications to line spectral estimation. *IEEE Transactions on Signal Processing*, 61(23):5987–5999, December 2013.

[38] S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning under covariate shift. *The Journal of Machine Learning Research*, 10:2137–2155, December 2009.

[39] S. Binsaeid, S. Asfour, S. Cho, and A. Onar. Machine ensemble approach for simultaneous detection of transient and gradual abnormalities in end milling using multisensor fusion. *Journal of Materials Processing Technology*, 209(10):4728–4738, June 2009.

[40] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer Publishing, Berlin Germany, 1st edition, 2006.

[41] J. Blough, D. Brown, and H. Vold. The time variant discrete fourier transform as an order tracking method. *SAE Technical Paper*, May 1997.

[42] D. A. J. Blythe, P. von Bünau, F. C. Meinecke, and K.-R. Müller. Feature extraction for change-point detection using stationary subspace analysis. *IEEE Transactions on Neural Networks and Learning Systems*, 23(4):631–643, April 2012.

[43] P. Borghesani, P. Pennacchi, S. Chatterton, and R. Ricci. The velocity synchronous discrete fourier transform for order tracking in the field of rotating machinery. *Mechanical Systems and Signal Processing*, 44(1–2):118–133, February 2014.

[44] D. C. Brabham. Crowdsourcing as a model for problem solving: An introduction and cases. *Convergence*, 14(1):75–90, February 2008.

[45] G. L. Bretthorst. *Bayesian Spectrum Analysis and Parameter Estimation*. Springer Publishing, Berlin Germany, 1st edition, 1988.

[46] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. LOF: Identifying density-based local outliers. In *Proc. of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD'00, pages 93–104, 2000.

[47] A. Bulling, U. Blanke, and B. Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys*, 46(3): 33:1–33:33, January 2014.

[48] A. S. L. O. Campanharo, M. I. Sirer, R. D. Malmgren, F. M. Ramos, and L. A. N. Amaral. Duality between time series and networks. *PloS one*, 6(8): e23378, August 2011.

[49] R. J. G. B. Campello, D. Moulavi, and J. Sander. Density-based clustering based on hierarchical density estimates. In *Advances in Knowledge Discovery and Data Mining*, PAKDD'13, pages 160–172, 2013.

[50] J. A. Carino, D. Zurita, M. Delgado, J. A. Ortega, and R. J. Romero-Troncoso. Remaining useful life estimation of ball bearings by means of monotonic score calibration. In *Proc. of the IEEE International Conference on Industrial Technology*, ICIT'15, pages 1752–1758, 2015.

[51] R. Carrière and R. L. Moses. High resolution radar target modeling using a modified prony estimator. *IEEE Transactions on Antennas and Propagation*, 40(1):13–18, January 1992.

[52] R. Chalapathy, A. K. Menon, and S. Chawla. Anomaly detection using one-class neural networks. *arXiv preprint arXiv:1802.06360*, January 2019.

[53] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):15:1–15:58, July 2009.

[54] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. The MIT Press, Cambridge USA, 1st edition, 2010.

[55] C. Chen, G. Vachtsevanos, and M. E. Orchard. Machine remaining useful life prediction: An integrated adaptive neuro-fuzzy and high-order particle filtering approach. *Mechanical Systems and Signal Processing*, 28:597–607, April 2012. Interdisciplinary and Integration Aspects in Structural Health Monitoring.

[56] H. Chen and N. Zhang. Graph-based change-point detection. *The Annals of Statistics*, 43(1):139–176, January 2015.

[57] N. Chen, Z.-S. Ye, Y. Xiang, and L. Zhang. Condition-based maintenance using the inverse gaussian degradation model. *European Journal of Operational Research*, 243(1):190–199, May 2015.

[58] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing*, EMNLP'14, pages 1724–1734, 2014.

[59] I. Cleland, M. Han, C. D. Nugent, H. Lee, S. I. McClean, S. Zhang, and S. Lee. Evaluation of prompted annotation of activity data recorded from a smart phone. *Sensors*, 14(9):15861–15879, August 2014.

[60] M. Costa and L. De Angelis. Model selection in hidden markov models: A simulation study. Technical report, Department of Statistics, University of Bologna, Bologna Italy, 2010.

[61] D. R. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological).*, 34(2):187—220, March 1972.

[62] F. Cruciani, I. Cleland, C. D. Nugent, P. J. McCullagh, K. Synnes, and J. Hallberg. Automatic annotation for human activity recognition in free living using a smartphone. *Sensors*, 18(7):2203–2222, July 2018.

[63] M. Cuturi and M. Blondel. Soft-DTW: A differentiable loss function for time-series. In *Proc. of the 34th International Conference on Machine Learning*, ICML'17, pages 894–903, 2017.

[64] P. Depalle, G. Garcia, and X. Rodet. Tracking of partials for additive sound synthesis using hidden markov models. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP'93, pages I–225–I–228, 1993.

[65] L. Dinh, D. Krueger, and Y. Bengio. NICE: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, April 2015.

[66] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real NVP. *arXiv preprint arXiv:1605.08803*, February 2017.

[67] Dittel Messtechnik GmbH. Balancing systems for grinding machines. Website, 2018.

[68] C. Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

[69] J. Dong, K. V. R. Subrahmanyam, Y. S. Wong, G. S. Hong, and A. R. Mohanty. Bayesian-inference-based neural networks for tool wear estimation. *The International Journal of Advanced Manufacturing Technology*, 30(9): 797–807, October 2006.

[70] S. Dong and T. Luo. Bearing degradation process prediction based on the PCA and optimized LS-SVM model. *Measurement*, 46(9):3143–3152, November 2013.

[71] L. Dou and R. J. W. Hodgson. Bayesian inference and gibbs sampling in spectral analysis and parameter estimation. i. *Inverse Problems*, 11:1069–1085, October 1995.

[72] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of the 2nd International Conference on Knowledge Discovery and Data Mining*, KDD'96, 1996.

[73] N. Esterer and P. Depalle. A linear programming approach to the tracking of partials. *arXiv preprint arXiv:1901.05044*, 2019.

[74] P. Fearnhead and G. Rigaill. Changepoint detection in the presence of outliers. *Journal of the American Statistical Association*, 113(525):1–15, 2018.

[75] C. Forbes, M. Evans, N. Hastings, and B. Peacock. *Statistical Distributions*. John Wiley & Sons, Hoboken USA, 4th edition, 2011.

[76] J. Foulds and E. Frank. A review of multi-instance learning assumptions. *The Knowledge Engineering Review*, 25(1):1–25, March 2010.

[77] F. Auger François and P. Flandrin. Improving the readability of time-frequency and time-scale representations by the reassignment method. *IEEE Transactions on Signal Processing*, 43(5):1068–1089, May 1995.

[78] T.-C. Fu. A review on time series data mining. *Engineering Applications of Artificial Intelligence*, 24:164–181, February 2011.

[79] K. R. Fyfe and E. D. S. Munck. Analysis of computed order tracking. *Mechanical Systems and Signal Processing*, 11(2):187–205, March 1997.

[80] R. Garnett, M. A. Osborne, and S. J. Roberts. Sequential bayesian prediction in the presence of changepoints. In *Proc. of the 26th International Conference on Machine Learning*, ICML'09, pages 345–352, 2009.

[81] T. Gerber, N. Martin, and C. Mailhes. Time-frequency tracking of spectral structures estimated by a data-driven method. *IEEE Transactions on Industrial Electronics*, 62(10):6616–6626, October 2015.

[82] N. Ghosh, Y. B. Ravi, A. Patra, S. Mukhopadhyay, S. Paul, A. R. Mohanty, and A. B. Chattopadhyay. Estimation of tool wear during CNC milling using neural network-based sensor fusion. *Mechanical Systems and Signal Processing*, 21(1):466–479, January 2007.

[83] A. Giantomassi, F. Ferracuti, A. Benini, G. Ippoliti, S. Longhi, and A. Petrucci. Hidden markov model for health estimation and prognosis of turbofan engines. In *Proc. of the ASME International Conference on Mechatronic and Embedded Systems and Applications*, pages 1–9, 2011.

[84] H. Gjoreski and D. Roggen. Unsupervised online activity discovery using temporal behaviour assumption. In *Proc. of the 2017 ACM International Symposium on Wearable Computers*, ISWC'17, 2017.

[85] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proc. of the 13th International Conference on Artificial Intelligence and Statistics*, AISTATS'10, 2010.

[86] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, David Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, NIPS'14, 2014.

[87] N. Görnitz, A. Porbadnigk, A. Binder, C. Sannelli, M. Braun, K.-R. Mueller, and M. Kloft. Learning and evaluation in presence of non-i.i.d. label noise. In *Proc. of the 17th International Conference on Artificial Intelligence and Statistics*, AISTATS'14, 2014.

[88] A. Goyal, A. Sordoni, M.-A. Côté, N. R. Ke, and Y. Bengio. Z-Forcing: Training stochastic recurrent networks. In *Advances in Neural Information Processing Systems 30*, NIPS'17, pages 6716–6726, 2017.

[89] L. Guo, N. Li, F. Jia, Y. Lei, and J. Lin. A recurrent neural network based health indicator for remaining useful life prediction of bearings. *Neurocomputing*, 240:98–109, May 2017.

[90] K. L. Gwet. *Handbook of inter-rater reliability: The definitive guide to measuring the extent of agreement among raters*. Advanced Analytics LLC, Gaithersburg USA, 4th edition, 2014.

[91] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Proc. of the 2006 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR'06, 2006.

[92] T. L. Hansen, B. H. Fleury, and B. D. Rao. Superfast line spectral estimation. *IEEE Transactions on Signal Processing*, 66(10):2511–2526, February 2018.

[93] Z. Harchaoui, E. Moulines, and F. R. Bach. Kernel change-point analysis. In *Advances in Neural Information Processing Systems 20*, NIPS'08, pages 609–616, 2009.

[94] V. Hautamaki, I. Karkkainen, and P. Franti. Outlier detection using k-nearest neighbour graph. In *Proc. of the 17th International Conference on Pattern Recognition*, ICPR'04.

[95] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR'16, 2016.

[96] Z. He, X. Xu, and S. Deng. Discovering cluster-based local outliers. *Pattern Recognition Letters*, 24(9-10):1641–1650, June 2003.

[97] U. Heisel, F. Klocke, E. Uhlmann, and G. Spur. *Handbuch Spanen*. Carl Hanser Verlag, Munich Germany, 2nd edition, 2016.

[98] J. Hernández-González, I. Inza, and J. A. Lozano. Learning bayesian network classifiers from label proportions. *Pattern Recognition*, 46(12):3425–3440, December 2013.

[99] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine*, 29(6):82–97, October 2012.

[100] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, November 2012.

[101] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, July 2006.

[102] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, November 1997.

[103] D. Hovy, T. Berg-Kirkpatrick, A. Vaswani, and E. Hovy. Learning whom to trust with MACE. In *Proc. of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics*, ACL'13, 2013.

[104] W.-N. Hsu and J. Glass. Scalable factorized hierarchical variational autoencoder training. *arXiv preprint arXiv:1804.03201*, 2018.

[105] W.-N. Hsu, Y. Zhang, and J. Glass. Learning latent representations for speech generation and transformation. In *Proc. of the 18th Annual Conference of the International Speech Communication Association*, Interspeech'17, 2017.

[106] W.-N. Hsu, Y. Zhang, and J. Glass. Unsupervised learning of disentangled and interpretable representations from sequential data. In *Advances in Neural Information Processing Systems 30*, pages 1878–1889, 2017.

[107] L. Hu, Z. Shi, J. Zhou, and Q. Fu. Compressed sensing of complex sinusoids: An approach based on dictionary refinement. *IEEE Transactions on signal processing*, 60(7):3809–3822, July 2012.

[108] R. Huang, L. Xi, X. Li, C. R. Liu, H. Qiu, and J. Lee. Residual life predictions for ball bearings based on self-organizing map and back propagation neural network methods. *Mechanical Systems and Signal Processing*, 21 (1):193–207, January 2007.

[109] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. of the 32nd International Conference on Machine Learning*, ICML'15, 2015.

[110] S. Jaganathan. Learning from weak label sources in industrial sensor systems. Master's thesis.

[111] A. K. S. Jardine, D. Lin, and D. Banjevic. A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, 20(7):1483–1510, October 2006.

[112] K. Jemielniak, R. Teti, J. Kossakowska, and T. Segreto. Innovative signal processing for cutting force based chip form prediction. In *Intelligent Production Machines and Systems*, 2nd I*PROMS Virtual International Conference, pages 7–12. Elsevier Science, Oxford UK, 2006.

[113] W. Jin, A. K. H. Tung, J. Han, and W. Wang. Ranking outliers using symmetric neighborhood relationship. In *Proc. of the 10th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, PAKDD'06, 2006.

[114] J.Pereira and M. Silveira. Unsupervised anomaly detection in energy time series data using variational recurrent autoencoders with attention. In *Proc. of the 2018 17th IEEE International Conference on Machine Learning and Applications*, ICMLA'18, pages 1275–1282, 2018.

[115] T. Kanamori, S. Hido, and M. Sugiyama. A least-squares approach to direct importance estimation. *The Journal of Machine Learning Research*, 10: 1391–1445, December 2009.

[116] Y. Kawahara, T. Yairi, and K. Machida. Change-point detection in time-series data based on subspace identification. In *Proc. of the 7th IEEE International Conference on Data Mining*, ICDM'07, pages 559–564, 2007.

[117] E. Keogh and C. A. Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7(3):358–386, March 2005.

[118] E. Keogh, S. Chu, D. Hart, and M. Pazzani. An online algorithm for segmenting time series. In *Proc. of the International Conference on Data Mining*, ICDM '01, pages 289–296, 2001.

[119] C. Kereliuk and P. Depalle. Improved hidden markov model partial tracking through time-frequency analysis. In *Proc. of the 11th International Conference on Digital Audio Effects*, DAFx'08, pages 1–4, 2008.

[120] J. P. Kharoufeh. Explicit results for wear processes in a markovian environment. *Operations Research Letters*, 31(3):237–244, May 2003.

[121] J. P. Kharoufeh, C. J. Solo, and M. Y. Ulukus. Semi-markov models for degradation-based reliability. *IIE Transactions*, 42(8):599–612, May 2010.

[122] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[123] D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems 31*, NeurIPS'18, 2018.

[124] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2014.

[125] Fritz Klocke. *Fertigungsverfahren 2*. Springer Publishing, Berlin Germany, 5th edition, 2017.

[126] J. Knoblauch, J. E. Jewson, and T. Damoulas. Doubly robust bayesian inference for non-stationary streaming data with β-divergences. In *Advances in Neural Information Processing Systems 31*, NIPS'18, pages 64–75, 2018.

[127] K. Kodera, C. De Villedary, and R. Gendrin. A new method for the numerical analysis of non-stationary signals. *Physics of the Earth and Planetary Interiors*, 12(2-3):142–150, August 1976.

[128] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek. LoOP: Local outlier probabilities. In *Proc. of the 18th ACM Conference on Information and Knowledge Management*, CIKM'09, 2009.

[129] H.-P. Kriegel, P. Kröger, and A. Zimek. Outlier detection techniques. In *Proc. of the SIAM International Conference on Data Mining*, SDM'10, pages 1–73, 2010. Tutorial.

[130] H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, February 1955.

[131] D. Kumar and B. Klefsjó. Proportional hazards model: a review. *Reliability Engineering and System Safety*, 44(2):177–188, 1994.

[132] A. Kusupati, M. Singh, K. Bhatia, A. Kumar, P. Jain, and M. Varma. FastGRNN: A fast, accurate, stable and tiny kilobyte sized gated recurrent neural network. In *Advances in Neural Information Processing Systems 31*, NIPS'18, pages 9031–9042, 2018.

[133] M. Lagrange, S. Marchand, M. Raspaud, and J.-B. Rault. Enhanced partial tracking using linear prediction. In *Proc. of the 6th International Conference on Digital Audio Effects*.

[134] G. Lai, B. Li, G. Zheng, and Y. Yang. Stochastic WaveNet: A generative latent variable model for sequential data. *arXiv preprint arXiv:1806.06116*, 2018.

[135] L. J. Latecki, A. Lazarevic, and D. Pokrajac. Outlier detection with kernel density functions. In *Proc. of the 5th International Conference on Machine Learning and Data Mining in Pattern Recognition*, MLDM'07, 2007.

[136] J. Lee, H. Qiu, G. Yu, J. Lin, and Rexnord Technical Services. Bearing data set. Website, 2007.

[137] A. W. Lees and M. I. Friswell. The evaluation of rotor imbalance in flexibly mounted machines. *Journal of Sound and Vibration*, 208(5):671–683, December 1997.

[138] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin. Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing*, 104:799–834, May 2018.

[139] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *Proc. of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'94, 1994.

[140] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology*, 49(4):764–766, July 2013.

[141] X. Li. A brief review: Acoustic emission method for tool wear monitoring during turning. *International Journal of Machine Tools and Manufacture*, 42(2):157–165, January 2002.

[142] X. Li, G. Ouyang, and Z. Liang. Complexity measure of motor current signals for tool flute breakage detection in end milling. *International Journal of Machine Tools and Manufacture*, 48(3):371–379, March 2008.

[143] T. W. Liao. Clustering of time series data - a survey. *Pattern Recognition*, 38(11):1857–1874, November 2005.

[144] Z. Liao, L. Song, P. Chen, Z. Guan, Z. Fang, and K. Li. An effective singular value selection and bearing fault signal filtering diagnosis method based on false nearest neighbors and statistical information criteria. *Sensors*, 18 (7):1–22, July 2018.

[145] D. Liu, J. Pang, J. Zhou, Y. Peng, and M. Pecht. Prognostics for state of health estimation of lithium-ion batteries based on combination gaussian process functional regression. *Microelectronics Reliability*, 53(6):832–839, June 2013.

[146] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *Proc. of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM'08, pages 413–422, 2008.

[147] H. Liu, L. Li, and J. Ma. Rolling bearing fault diagnosis based on STFT-deep learning and sound signals. *Shock and Vibration*, 2016:1–12, July 2016.

[148] Q. Liu and H.-P. Wang. A case study on multisensor data fusion for imbalance diagnosis of rotating machinery. *Artificial intelligence for engineering design analysis and manufacturing*, 15(3):203–210, June 2001.

[149] R. Liu, B. Yang, E. Zio, and X. Chen. Artificial intelligence for fault diagnosis of rotating machinery: A review. *Mechanical Systems and Signal Processing*, 108:33–47, August 2018.

[150] S. Liu, M. Yamada, N. Collier, and M. Sugiyama. Change-point detection in time-series data by relative density-ratio estimation. *Neural Networks*, 43:72–83, July 2013.

[151] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30*, NIPS'18, pages 4765–4774, 2017.

[152] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. of the 30th International Conference on Machine Learning*, ICML'13, 2015.

[153] A. Makhzani and B. Frey. k-Sparse autoencoders. *arXiv preprint arXiv:1312.5663*, December 2013.

[154] A. Malhi, R. Yan, and R. X. Gao. Prognosis of defect propagation based on recurrent neural networks. *IEEE Transactions on Instrumentation and Measurement*, 60(3):703–711, February 2011.

[155] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal. Long short term memory networks for anomaly detection in time series. In *Proc. of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, ESANN'15, 2015.

[156] D. Malioutov, M. Cetin, and A. S. Willsky. A sparse signal reconstruction perspective for source localization with sensor arrays. *IEEE Transactions on signal processing*, 53(8):3010–3022, August 2005.

[157] E. Manzoor, H. Lamba, and L. Akoglu. xStream: Outlier detection in feature-evolving data streams. In *Proc. of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD'18, 2018.

[158] O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. In *Advances in Neural Information Processing Systems 10*, NIPS'98, pages 570–576, 1998.

[159] A. Maslov, M. Pechenizkiy, Y. Pel, I. Žliobaitė, A. Shklyaev, T. Karkkäinen, and J. Hollmén. BLPA: Bayesian learn-predict-adjust method for online detection of recurrent changepoints. In *Proc. of the 2017 International Joint Conference on Neural Networks*, IJCNN'17, pages 1916–1923, 2017.

[160] R. R. McAulay and T. F. Quatieri. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(4):744–754, August 1986.

[161] N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436–1462, June 2006.

[162] S. Mills, T. P. Pridmore, and M. Hills. Tracking in a hough space with the extended kalman filter. In *Proc. of the 14th British Machine Vision Conference*, BMVC'03, pages 18.1–18.10, 2003.

[163] B. D. Minor. Toward learning and mining from uncertain time-series data for activity prediction. In *SIGKDD Workshop on Mining and Learning from Time Series*, MiLeTS'15, pages 1–10, 2015.

[164] T. Miu, T. Plötz, P. Missier, and D. Roggen. On strategies for budget-based online annotation in human activity recognition. In *Proc. of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, UbiComp '14 Adjunct, 2014.

[165] T. Miu, P. Missier, and T. Plötz. Bootstrapping personalised human activity recognition models using online active learning. In *Proc. of the 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, CIT'15, 2015.

[166] P. G. Moreno, A. Artés-Rodríguez, Y. W. Teh, and F. Perez-Cruz. Bayesian nonparametric crowdsourcing. *Journal of Machine Learning Research*, 16: 1607–1627, January 2015.

[167] K. P. Murphy. Conjugate bayesian analysis of the gaussian distribution. Technical report, University of British Columbia, Vancouver Canada, 2007.

[168] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, Cambridge, USA, 1st edition, 2012.

[169] P. Nectoux, R. Gouriveau, K. Medjaher, E. Ramasso, B. Chebel-Morello, N. Zerhouni, and C. Varnier. PRONOSTIA: An experimental platform for bearings accelerated life test. In *Prof. of the IEEE International Conference on Prognostics and Health Management*, PHM'12, 2012.

[170] J. Neri. Fast partial tracking of audio with real-time capability through linear programming. https://github.com/jundsp/Fast-Partial-Tracking. Accessed: June 27, 2019.

[171] J. Neri and P. Depalle. Fast partial tracking of audio with real-time capability through linear programming. In *Proc. of the 21st International Conference on Digital Audio Effects*, DAFx'18, pages 326–333, 2018.

[172] C. Nicolaou. Statistische Modelle zur Segmentierung rekurrenter Signale in Industrie 4.0 Anwendungen. Master's thesis.

[173] A. Niculescu-Mizil and R. Caruana. Predicting good probabilities with supervised learning. In *Proc. of the 22nd International Conference on Machine Learning*, ICML'05, pages 625–632, 2005.

[174] H. Ocak, K. A. Loparo, and F. M. Discenzo. Online tracking of bearing wear using wavelet packet decomposition and probabilistic modeling: A method for bearing prognostics. *Journal of Sound and Vibration*, 302(4): 951–961, May 2007.

[175] J. F. G. Oliveira and D. A. Dornfeld. Application of AE contact sensing in reliable grinding monitoring. *CIRP Annals-Manufacturing Technology*, 50 (1):217–220, July 2001.

[176] J. F. G. Oliveira, E. J. Silva, C. Guo, and F. Hashimoto. Industrial challenges in grinding. *CIRP Annals-Manufacturing Technology*, 58(2):663–680, August 2009.

[177] A. Van Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. In *Proc. of the 33rd International Conference on Machine Learning*, ICML'16.

[178] B. Ottersten, M. Viberg, and T. Kailath. Analysis of subspace fitting and ML techniques for parameter estimation from sensor array data. *IEEE Transactions on signal processing*, 40(3):590–600, March 1992.

[179] A. Y. Ouadine, M. Mjahed, H. Ayad, and A. El Kari. Helicopter gearbox vibration fault classification using order tracking method and genetic algorithm. *Automatika*, 60(1):68–78, February 2019.

[180] E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, June 1954.

[181] J. Paparrizos and L. Gravano. k-Shape: Efficient and accurate clustering of time series. *SIGMOD Record*, 45(1):69–76, June 2016.

[182] R. J. Passonneau and B. Carpenter. The benefits of a model of annotation. *Transactions of the Association for Computational Linguistics*, 2:311–326, October 2014.

[183] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. In *Advances in Neural Information Processing Systems 30*, NIPS'17, pages 1–4, 2017. NIPS Autodiff Workshop.

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

[185] Y. Peng. Empirical model decomposition based time-frequency analysis for the effective detection of tool breakage. *Journal of Manufacturing Science and Engineering*, 128(1):154–166, September 2004.

[186] F. Petitjean, A. Ketterlin, and P. Gançarski. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 44(3):678–693, March 2011.

[187] F. Petitjean, G. Forestier, G. I. Webb, A. E. Nicholson, Y. Chen, and E. Keogh. Faster and more accurate classification of time series by exploiting a novel dynamic time warping averaging algorithm. *Knowledge and Information Systems*, 47(1):1–26, April 2016.

[188] T. Pevny. Loda: Lightweight on-line detector of anomalies. *Machine Learning*, 102(2):275–304, February 2016.

[189] T. Plötz, C. Chen, N. Y. Hammerla, and G. D. Abowd. Automatic synchronization of wearable sensors and video-cameras for ground truth annotation – a practical approach. In *Proc. of the 2012 16th Annual International Symposium on Wearable Computers*, ISWC'12, pages 100–103, 2012.

[190] L. Prechelt. *Early Stopping — But When?*, pages 53–67. Springer Publishing, Berlin Germany, 2012.

[191] P. W. Prickett and C. Johns. An overview of approaches to end milling tool monitoring. *International Journal of Machine Tools and Manufacture*, 39 (1):105–122, January 1999.

[192] Y. Qian, R. Yan, and S. Hu. Bearing degradation evaluation using recurrence quantification analysis and kalman filter. *IEEE Transactions on Instrumentation and Measurement*, 63(11):2599–2610, November 2014.

[193] H. Qiu, J. Lee, J. Lin, and G. Yu. Robust performance degradation assessment methods for enhanced rolling element bearing prognostics. *Advanced Engineering Informatics*, 17(3):127–140, July 2003. Intelligent Maintenance Systems.

[194] N. Quadrianto, A. J. Smola, T. S. Caetano, and Q. V. Le. Estimating labels from label proportions. *Journal of Machine Learning Research*, 10:2349–2374, December 2009.

[195] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77, pages 267–296. Morgan Kaufmann Publishers Inc., San Francisco USA, February 1990.

[196] A. J. Ratner, C. M. De Sa, S. Wu, D. Selsam, and C. Ré. Data programming: Creating large training sets, quickly. In *Advances in Neural Information Processing Systems 29*, NIPS'16, pages 3567–3575, 2016.

[197] P. Ravikumar, M. J. Wainwright, and J. D. Lafferty. High-dimensional ising model selection using ℓ_1-regularized logistic regression. *The Annals of Statistics*, 38(3):1287–1319, June 2010.

[198] S. Rayana and L. Akoglu. Less is more: Building selective anomaly ensembles. *ACM Transactions on Knowledge Discovery from Data*, 10(4): 42:1–42:33, May 2016.

[199] S. Rayana, W. Zhong, and L. Akoglu. Sequential ensemble learning for outlier detection: A bias-variance perspective. In *Prof. of the IEEE 16th International Conference on Data Mining*, ICDM'16, pages 1167–1172, 2016.

[200] C. Reich, A. Mansour, and K. Van Laerhoven. Embedding intelligent features for vibration-based machine condition monitoring. In *Proc. of the IEEE 26th European Signal Processing Conference*, EUSIPCO'18, pages 371–375, 2018.

[201] C. Reich, A. Mansour, and K. Van Laerhoven. Collecting labels for rare anomalies via direct human feedback — an industrial application study. *Informatics*, 6(3):article nr. 38, September 2019.

[202] C. Reich, C. Nicolaou, A. Mansour, and K. Van Laerhoven. Bayesian estimation of recurrent changepoints for signal segmentation and anomaly detection. In *Proc. of the IEEE 27th European Signal Processing Conference*, EUSIPCO'19, pages 1–5, 2019.

[203] C. Reich, C. Nicolaou, A. Mansour, and K. Van Laerhoven. Detection of machine tool anomalies from bayesian changepoint recurrence estimation. In *Proc. of the IEEE 17th International Conference on Industrial Informatics*, INDIN'19, pages 1297–1302, 2019.

[204] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proc. of the 28th International Conference on Machine Learning*, ICML'11, 2011.

[205] P. J. Rousseeuw and K. Van Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, August 1999.

[206] R. Roy and T. Kailath. ESPRIT – estimation of signal parameters via rotational invariance techniques. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(7):369–411, July 1990.

[207] L. Ruff, R. Vandermeulen, N. Görnitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft. Deep one-class classification. In *Proc. of the 35th International Conference on Machine Learning*, ICML'18, 2018.

[208] L. Ruff, R. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, and M. Kloft. Deep Semi-Supervised Anomaly Detection. *arXiv preprint arXiv:1906.02694*, June 2019.

[209] Y. Saatçi, R. D. Turner, and C. E. Rasmussen. Gaussian process change point models. In *Proc. of the 27th International Conference on Machine Learning*, ICML'10, pages 927–934, 2010.

[210] D. R. Salgado and F. J. Alonso. Tool wear detection in turning operations using singular spectrum analysis. *Journal of Materials Processing Technology*, 171(3):451–458, February 2006.

[211] D. R. Salgado, F. J. Alonso, I. Cambero, and A. Marcelo. In-process surface roughness prediction system using cutting vibrations in turning. *The International Journal of Advanced Manufacturing Technology*, 43(1):40–51, July 2009.

[212] A. Samé, F. Chamroukhi, G. Govaert, and P. Aknin. Model-based clustering and segmentation of time series with changes in regime. *Advances in Data Analysis and Classification*, 5(4):301–321, December 2013.

[213] A. Sapena-Bano, J. Burriel-Valencia, M. Pineda-Sanchez, R. Puche-Panadero, and M. Riera-Guasp. The harmonic order tracking analysis method for the fault diagnosis in induction motors under time-varying conditions. *IEEE Transactions on Energy Conversion*, 32(1):244–256, March 2017.

[214] H. Satar-Boroujeni and B. Shafai. A robust algorithm for partial tracking of music signals. In *Proc. of the 8th International Conference on Digital Audio Effects*.

[215] S. Sathe and C. C. Aggarwal. Subspace outlier detection in linear time with randomized hashing. In *Proc. of the IEEE 16th International Conference on Data Mining*, ICDM'16, pages 459–468, 2016.

[216] Schaudt Mikrosa GmbH. KRONOS S. Präzision für kleine Werkstücke.

[217] C. Scheffer and P. S. Heyns. Wear monitoring in turning operations using vibration and strain measurements. *Mechanical Systems and Signal Processing*, 15(6):1185–1202, November 2001.

[218] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *Information Processing in Medical Imaging*, pages 146–157, 2017.

[219] R. O. Schmidt. Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation*, 34(3):276–280, March 1986.

[220] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computing*, 13(7):1443–1471, July 2001.

[221] M. Schröder, K. Yordanova, S. Bader, and T. Kirste. Tool support for the online annotation of sensor data. In *Proc. of the 3rd International Workshop on Sensor-based Activity Recognition and Interaction*, iWOAR '16, 2016.

[222] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A unified embedding for face recognition and clustering. In *Proc. of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR'15, 2015.

[223] E. Schubert, A. Zimek, and H.-P. Kriegel. Generalized outlier detection with flexible kernel density estimates. In *Proc. of the 2014 SIAM International Conference on Data Mining*, pages 542–550, 2014.

[224] E. Schubert, A. Zimek, and H.-P. Kriegel. Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection. *Data Mining and Knowledge Discovery*, 28(1): 190–237, January 2014.

[225] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, March 1978.

[226] B. Settles. Active learning literature survey. Technical report, Department of Computer Science, University of Wisconsin–Madison, Madison USA, January 2010.

[227] Y.-T. Sheen and C.-K. Hung. Constructing a wavelet-based envelope function for vibration signal analysis. *Mechanical Systems and Signal Processing*, 18(1):119–126, January 2004.

[228] D. Shi and N. N. Gindy. Tool wear predictive model based on least squares support vector machines. *Mechanical Systems and Signal Processing*, 21 (4):1799–1814, May 2007.

[229] X.-S. Si, W. Wang, C.-H. Hu, D.-H. Zhou, and M. G. Pecht. Remaining useful life estimation based on a nonlinear diffusion degradation process. *IEEE Transactions on Reliability*, 61(1):50–67, March 2012.

[230] B. Sick. On-line and indirect tool wear monitoring in turning with artificial neural networks: A review of more than a decade of research. *Mechanical Systems and Signal Processing*, 16(4):487–546, July 2002.

[231] S. M. Siddiqi, G. J. Gordon, and A. W. Moore. Fast state discovery for HMM model selection and learning. In *Proc. of the 11th International Conference on Artificial Intelligence and Statistics*, AISTATS'07, pages 492–499, 2007.

[232] S. M. Siddiqi, G. J. Gordon, and A. W. Moore. Fast state discovery for HMM model selection and learning, 2007.

[233] I. Siegert, R. Böck, and A. Wendemuth. Inter-rater reliability for emotion annotation in human–computer interaction: comparison and methodological improvements. *Journal on Multimodal User Interfaces*, 8(1):17–28, March 2014.

[234] J. K. Sinha, A. W. Lees, and M. I. Friswell. Estimating unbalance and misalignment of a flexible rotating machine from a single run-down. *Journal of Sound and Vibration*, 272(3–5):967–989, May 2004.

[235] F. Sloukia, M. El Aroussi, H. Medromi, and M. Wahbi. Bearings prognostic using mixture of gaussians hidden markov model and support vector machine. In *Proc. of the ACS International Conference on Computer Systems and Applications*, AICCSA'13, pages 1–4, 2013.

[236] S. L. Smith, P.-J. Kindermans, C. Ying, and Q. V. Le. Don't decay the learning rate, increase the batch size. *arXiv preprint arXiv:1711.00489*, 2017.

[237] R. Snow, B. O'Connor, D. Jurafsky, and A. Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proc. of the conference on empirical methods in natural language processing*, pages 254–263, 2008.

[238] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, January 2014.

[239] P. Stoica and R. Moses. *Spectral Analysis of Signals*. Prentice-Hall, Upper Saddle River USA, 1st edition, 2005.

[240] P. Stoica, P. Babu, and J. Li. SPICE: A sparse covariance-based estimation method for array processing. *IEEE Transactions on Signal Processing*, 59 (2):629–638, February 2011.

[241] K. C. S. Stoll and K. Otto. *Manual. Centerless External Cylindrical Grinding.* Schaudt Mikrosa GmbH, 2016.

[242] A. Storkey. *When Training and Test Sets are Different: Characterising Learning Transfer*, pages 3–28. The MIT Press, Cambridge USA, 2008.

[243] D. Stowell, S. Muševič, J. Bonada, and M. D. Plumbley. Improved multiple birdsong tracking with distribution derivative method and markov renewal process clustering. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP'13, pages 468–472, 2013.

[244] M. Sugiyama, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Proc. of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, pages 1433–1440, 2007.

[245] J. Sun, G. S. Hong, M. Rahman, and Y. S. Wong. Identification of feature set for effective tool condition monitoring by acoustic emission sensing. *International Journal of Production Research*, 42(5):901–918, February 2004.

[246] J.-I. Takeuchi and K. Yamanishi. A unifying framework for detecting outliers and change points from time series. *IEEE transactions on Knowledge and Data Engineering*, 18(4):482–492, April 2006.

[247] S. C. Tan, K. M. Ting, and T. F. Liu. Fast anomaly detection for streaming data. In *Proc. of the 22nd International Joint Conference on Artificial Intelligence*, IJCAI'11, 2011.

[248] J. Tang, Z. Chen, A. W.-C. Fu, and D. W.-L. Cheung. Enhancing effectiveness of outlier detections for low density patterns. In *Proc. of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, PAKDD'02, pages 535–548, 2002.

[249] D. M. J. Tax and R. P. W. Duin. Support vector data description. *Machine Learning*, 54(1):45–66, January 2004.

[250] D. M. J. Tax, A. Ypma, and R. P. W. Duin. Support vector data description applied to machine vibration analysis. In *Proc. of the 5th Annual Conference of the Advanced School for Computing and Imaging*, volume 54, pages 15–23, 1999.

[251] R. Teti, I.S. Jawahir, K. Jemielniak, T. Segreto, S. Chen, and J. Kossakowska. Chip form monitoring through advanced processing of cutting force sensor signals. *CIRP Annals*, 55(1):75–80, 2006.

[252] R. Teti, K. Jemielniak, G. O'Donnell, and D. Dornfeld. Advanced monitoring of machining operations. *CIRP Annals-Manufacturing Technology*, 59 (2):607–822, August 2010.

[253] R. Turner, Y. Saatçi, and C. E. Rasmussen. Adaptive sequential bayesian change point detection. Technical report, University of Cambridge, Cambridge UK, 2009.

[254] Y. Vaizman, K. Ellis, G. Lanckriet, and N. Weibel. ExtraSensory app: Data collection in-the-wild with rich user interface to self-report behavior. In *Proc. of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI'18, 2018.

[255] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, September 2016.

[256] J. M. van Noortwijk. A survey of the application of gamma processes in maintenance. *Reliability Engineering and System Safety*, 94(1):2–21, January 2009.

[257] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proc. of the 25th International Conference on International Conference on Machine Learning*.

[258] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11:3371–3408, December 2010.

[259] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11:3371–3408, December 2010.

[260] H. Vold and J. Leuridan. High resolution order tracking at extreme slew rates using kalman tracking filters. *Shock and Vibration*, 2(6), May 1995.

[261] M. Vrigkas, C. Nikou, and I. A. Kakadiaris. A review of human activity recognition methods. *Frontiers in Robotics and AI*, 2:28, November 2015.

[262] W. Q. Wang, M. F. Golnaraghi, and F. Ismail. Prognosis of machine health condition using neuro-fuzzy systems. *Mechanical Systems and Signal Processing*, 18(4):813–831, July 2004.

[263] X. Wang and D. Xu. An inverse gaussian process model for degradation data. *Technometrics*, 52(2):188–197, May 2010.

[264] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh. Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, 26(2): 275–309, March 2013.

[265] Z. Wang and T. Oates. Encoding time series as images for visual inspection and classification using tiled convolutional neural networks. In *Trajectory-Based Behavior Analytics: Papers from the 2015 AAAI Workshop*, pages 40–46, 2015.

[266] Z. Wang, W. Yan, and T. Oates. Time series classification from scratch with deep neural networks: A strong baseline. In *Proc. of the 2017 International Joint Conference on Neural Networks*, IJCNN'17, pages 1578–1585, 2017.

[267] X. Wen and M. Sandler. Notes on model-based non-stationary sinusoid estimation methods using derivatives. In *Proc. of the 12th International Conference on Digital Audio Effects*.

[268] A. Willsky and H. Jones. A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems. *IEEE Transactions on Automatic control*, 21(1):108–112, February 1976.

[269] R. C. Wilson, M. R. Nassar, and J. I. Gold. Bayesian online learning of the hazard rate in change-point problems. *Neural Computation*, 22(9):2452–2476, September 2010.

[270] P. Wirfält, G. Bouleux, M. Jansson, and P. Stoica. Subspace-based frequency estimation utilizing prior information. In *Proc. of the Statistical Signal Processing Workshop*, SSP'11, pages 533–536, 2011.

[271] F. Wu and L. Qu. Diagnosis of subharmonic faults of large rotating machinery based on EMD. *Mechanical Systems and Signal Processing*, 23(2): 467–475, February 2009.

[272] K. Wu, K. Zhang, W. Fan, A. Edwards, and P. S. Yu. RS-forest: A rapid density estimator for streaming anomaly detection. In *Proc. of the 2014 IEEE International Conference on Data Mining*, ICDM'14, pages 600–609, 2014.

[273] Y. Wu and R. Du. Feature extraction and assessment using wavelet packets for monitoring of machining processes. *Mechanical Systems and Signal Processing*, 10(1):29–53, January 1996.

[274] X. Xi, E. Keogh, C. Shelton, L. Wei, and C. A. Ratanamahatana. Fast time series classification using numerosity reduction. In *Proc. of the 23rd International Conference on Machine Learning*, ICML'06, 2006.

[275] H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, y. Feng, J. Chen, Z. Wang, and H. Qiao. Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications. In *Proc. of the 2018 World Wide Web Conference on World Wide Web*, WWW'18, pages 187–196, 2018.

[276] M. Yamada, T. Suzuki, T. Kanamori, H. Hachiya, and M. Sugiyama. Relative density-ratio estimation for robust distribution comparison. In *Proc. of the*

24th International Conference on Neural Information Processing Systems, NIPS'11, pages 594–602, 2011.

[277] M. Yamada, A. Kimura, F. Naya, and H. Sawada. Change-point detection with feature selection in high-dimensional time-series data. In *Proc. of the 23rd International Joint Conference on Artificial Intelligence*, IJCAI'13, pages 1827–1833, 2013.

[278] W. Yang and P. J. Tavner. Empirical mode decomposition, an adaptive approach for interpreting shaft vibratory signals of large rotating machinery. *Journal of Sound and Vibration*, 321(3–5):1144–1170, April 2009.

[279] Y. Yang, M. Zhang, W. Chen, W. Zhang, H. Wang, and M. Zhang. Adversarial learning for chinese NER from crowd annotations. In *Proc. of the 32nd AAAI Conference on Artificial Intelligence*, AAAI'18, pages 1627–1634, 2018.

[280] Z. Yang and L. Xie. On gridless sparse methods for line spectral estimation from complete and incomplete data. *IEEE Transactions on signal processing*, 63(12):3139–3153, April 2015.

[281] Z. Yang and L. Xie. A weighted atomic norm approach to spectral superresolution with probabilistic priors. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP'16, pages 4598–4602, 2016.

[282] S.-Z. Yu. Hidden semi-markov models. *Artificial Intelligence*, 174(2):215–243, February 2010.

[283] D. Zachariah, P. Wirfält, M. Jansson, and S. Chatterjee. Line spectrum estimation with probabilistic priors. *Signal Processing*, 93(11):2969–2974, September 2013.

[284] A. Zenonos, A. Khan, G. Kalogridis, S. Vatsikas, T. Lewis, and M. Sooriyabandara. HealthyOffice: Mood recognition at work using smartphones and wearable sensors. In *Proc. of the IEEE International Conference on Pervasive Computing and Communication Workshops*, PerCom'16 Workshops, pages 1–6, 2016.

[285] C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, and N. V. Chawla. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. *arXiv preprint arXiv:1811.08055*, November 2018.

[286] K. Zhang, M. Hutter, and H. Jin. A new local distance-based outlier detection approach for scattered real-world data. In *Proc. of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, PAKDD'09, 2009.

[287] Q. Zhao, V. Hautamaki, and P. Fränti. Knee point detection in BIC for detecting the number of clusters. In *Proc. of the 10th International Conference on Advanced Concepts for Intelligent Vision Systems*, ACIVS'08, pages 664–673, 2008.

[288] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R.X. Gao. Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, 115:213–237, January 2019.

[289] X. Zhao, Q. Kong, and Q. Guo. Study of time-frequency order tracking of vibration signals of rotating machinery in changing state. In *IEEE International Symposiums on Information Processing*, ISIP'08, pages 559–563, 2008.

[290] Y. Zhao, Z. Nasrullah, and Z. Li. PyOD: A python toolbox for scalable outlier detection. *arXiv preprint arXiv:1901.01588*, January 2019.

[291] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proc. of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR'15, pages 2921–2929, 2015.

[292] Z.-H. Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, January 2018.

[293] J. Zhu, Q. Zhang, P. Gerstoft, M.-A. Badiu, and Z. Xu. Grid-less variational bayesian line spectral estimation with multiple measurement vectors. *Signal Processing*, 161:155–164, August 2019.

[294] J. Zhu, Q. Zhang, and X. Men. Grid-less variational bayesian inference of line spectral estimation from quantized samples. Technical report, Zhejiang University, Hangzhou China, 2019.

[295] X. Zhu. Semi-supervised learning literature surve. Technical report, Department of Computer Science, University of Wisconsin–Madison, Madison USA, July 2008.

[296] A. Zimek, M. Gaudet, R. J.G.B. Campello, and J. Sander. Subsampling for efficient and effective unsupervised outlier detection ensembles. In *Proc. of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD'13, 2013.

[297] A. Zimek, R. J.G.B. Campello, and J. Sander. Ensembles for unsupervised outlier detection: Challenges and research questions a position paper. *SIGKDD Exploration Newsletters*, 15(1):11–22, March 2014.

[298] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *Proc. of the 6th International Conference on Learning Representations*, ICLR'18, pages 833–840, 2018.