Dipl.-Ing. Stefan Schrunner, B.Sc.

Pattern Recognition in Analog Wafer Test Data

A Health Factor for Process Patterns

DOCTORAL THESIS

to achieve the university degree of

Doktor der technischen Wissenschaften

submitted to

Graz University of Technology

Supervisor
Univ.-Prof. Dr. Stefanie Lindstaedt

Institute of Interactive Systems and Data Science
Faculty of Computer Science and Biomedical Engineering

Advisor
Dr. Roman Kern

Graz, August 2019

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present doctoral thesis.

27.08.2019 Dipl.-Ing. Stefan Schrunner, B.Sc.

Data is the new oil. The companies that will win are using math.
(Kevin Plank, Founder and CEO of Under Armour, 2016)

Acknowledgements

During my work on this thesis[1], many people supported me from technical, organizational and personal perspective. At this point, I would like to express my gratitude to them.

First, I would like to thank my advisors at KAI Kompetenzzentrum Automobil- und Industrieelektronik GmbH, Dr. Olivia Pfeiler and Dr. Anja Zernig, as well as my scientific advisor, Dr. Roman Kern (Know-Center GmbH and Graz University of Technology). They provided me with valuable inputs, support and feedback at all stages of my PhD studies and encouraged me to develop my own approaches and ideas. Further, Dr. Andre Kästner (Infineon Technologies Austria AG) provided me with the necessary insights into the semiconductor industry and was always ready to answer any technical question. Thank you very much.

Furthermore, I am particularly grateful to my colleagues within the SemI40 project group at KAI GmbH, who contributed a lot to my ideas and approaches. In particular, DI Michael Scheiber gave me a lot of support with improving, enhancing and bug-fixing the implementations and conducting evaluations. Furthermore, three Master theses authored by DI Anna Jenul, DI Martin Pleschberger and DI Vedo Alagić accompanied my work and contributed a lot to the success of this project. In general, the mathematics/statistics/data science team at KAI GmbH provided me with valuable ideas in fruitful discussions.

I would like to express my special thanks to Dr. Bernhard C. Geiger and DI Tiago Teixeira dos Santos, who contributed not only as cooperation partners from Know-Center GmbH within our joint use-case, but also inspired me with their profound know-how. I particularly appreciated Bernhard's critical feedback on our joint publications.

Thanks also to DI Josef Fugger, CEO of KAI GmbH, who gave me the opportunity to work on this interesting topic in a very pleasant working environment. Further, I would like to thank my supervisor, Prof. Dr. Stefanie Lindstaedt, and my external reviewer, Prof. Dr. Jürgen Pilz. Both of them immediately agreed to review this PhD thesis and serve as examiners for the final PhD defense.

Last but not least, I want to express my special thanks to my girlfriend Anna, who contributed to this work with technical inputs and with her personal support, as well as to my parents. They encouraged me at any time to stay focused on my goals and were always there when I needed them. Further, I would like to thank my family and friends, in particular my colleagues at the Austrian Water Rescue Federation (Österreichische Wasserrettung), who supported me mentally throughout the 3 years of my PhD studies. Without all of these people, it would not have been possible for me to complete this thesis.

[1] The work has been performed in the project Power Semiconductor and Electronics Manufacturing 4.0 (SemI40), under grant agreement No 692466. The project is co-funded by grants from Austria (BMVIT-IKT der Zukunft, FFG project no. 853338), Germany, Italy, France, Portugal and - Electronic Component Systems for European Leadership Joint Undertaking (ECSEL JU).

Abstract

Semiconductor manufacturing is a highly complex and competitive branch of industry, comprising hundreds of process steps, which do not allow any deviations from the specification. Depending on the application area of the products, the production chain is subject to strict quality requirements. While heading towards industry 4.0, automation of production workflows is required and hence, even more effort must be spent on controlling the processes accordingly. The need for data-driven indicators supporting human experts via monitoring the production process is inevitable, but lacks adequate solutions exploiting both, profound academic methodologies and domain-specific know-how.

In many cases, process deviations cannot be detected automatically during the semiconductor frontend production. Hence, the wafer test stage at the end of frontend manufacturing plays a key role to determine whether preceding process steps were executed with the necessary precision. The analysis of these wafer test data is challenging, since process deviations can only be detected by investigating spatial dependencies (patterns) over the wafer. Such patterns become visible, if devices on the wafer violate specification limits of the product. In this work, we go one step further and investigate the automated detection of process patterns in data from analog wafer test parameters, i.e. the electrical measurements, instead of pass/fail classifications, which brings the benefit that deviations can be recognized before they result in yield loss - this aspect is a clear difference to state-of-the-art research, where merely specification violations are observed. For this purpose, an indicator for the level of concern associated with process patterns on the wafer, a so-called Health Factor for Process Patterns, is presented. The indicator combines machine learning techniques and expert knowledge.

In order to develop such a Health Factor, the problem is divided into three major components, which are investigated separately: recognition of the pattern type, quantification of the intensity of a pattern and specification of the criticality associated with each pattern type. Since the first two components are intrinsically present in the wafer test data, machine learning systems are deployed for both, while criticality is specified by introducing expert and domain knowledge to the concept. The proposed decision support system is semi-automated and thus, unifies pattern recognition and expert knowledge in a promising way.

The effectiveness of the proposed Health Factor is underlined by experiments conducted on simulated as well as real-world datasets. The evaluations show that the system is not only mathematically valid, but also practically applicable and fulfills the demands raised by a real-world production environment. Moreover, the indicator can be transferred to various product types or even related problem setups given a reliable training dataset.

Keywords: pattern recognition, wafer test data, decision support system, process monitoring, machine learning, industry 4.0, Bayesian decision theory

Contents

1 Introduction

1.1 Motivation

Historically, an era called *industrial revolution* started in the mid 18th century, initiated by the need to produce in an increasingly efficient way. Triggered by major inventions, the process started from rudimentary hand production, passed several intermediate stages, including assembly lines and mass production, and finally arrived at a state of digitalization today. However, apart from these achievements of the so-called industrial revolutions 1 to 3, yet not all possibilities of human inventions have been exploited - resulting in a 4th industrial revolution (industry 4.0) [1], driven by buzzwords such as cyber-physical systems [2], internet of things [3], nanotechnology or quantum computing, which dominate the industrial and academic world of the 2010s (and will probably continue to do so in the 2020s). The clear goal is to obtain fully-automated, intelligent production systems, while reducing the need for human interaction during manufacturing and, in case that a human expert is required, to provide as much insight as possible to such a person by data-driven tools. A brief overview on the major characteristics of the industrial revolutions is provided in Fig. 1.1, additionally stating key achievements at each stage.

Semiconductor fabrication is a highly competitive field, involving challenges from technical as well as from business side. Due to the growth of the whole electronics market and the ubiquitous use for high-performance hardware, the need for sensors, microcontrollers, etc. has increased enormously. Especially automation in all aspects of human life, such as smart home, smart city, smart factory, as well as autonomous driving are main key factors for this development. Statistical forecasts by WSTS Inc. [5] and Statista Inc. [6] agree on the prognosis that worldwide semiconductor industry sales revenue will exceed a limit of 490 billion USD, rapidly increasing each year. Hence, this value has doubled within the last decade.

Semiconductors are an essential part of the process towards fully-automated industrial systems. In particular, they are related to automation in two separate aspects: on the one hand, industry electronics and decision-making systems require them as major components, e.g. recording substantial sensor data to monitor the process and detect anomalies or deviations. In fact, the number of semiconductors (sensors, microcontrollers, etc.) to control any fully-automated system is in a very high range, which is in particular apparent by the high effort the semiconductor industry dedicates to the development of self-driving cars.

On the other hand, semiconductor manufacturing itself is highly complex and a prime example for the potential of automated production and process control. This fact is underlined by a huge amount of gathered data from various sources, used in order to thoroughly monitor each single process step. Such data volume is unmanageable by human experts, but rather fits the capabilities of machines to perform scalable data analysis. Especially process deviations are harmful

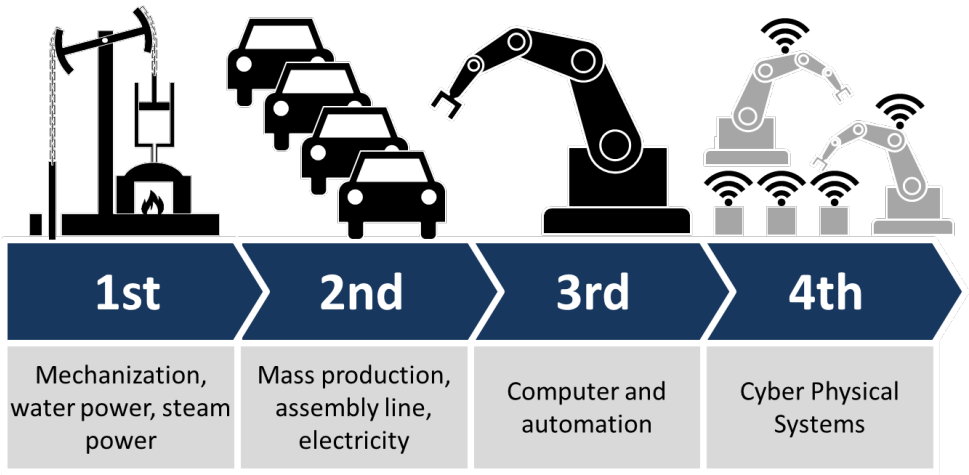


Fig. 1.1: The industrial revolution, originally started in the mid 18th century, has passed 4 major phases up to now. Today, we face the 4th industrial revolution and its challenges. [4]

during semiconductor manufacturing as high precision is required and complex interactions between physical and chemical processes must be controlled. Hence, process control is a crucial target for data analysis, which is still insufficiently covered by data-driven tools. Although experts monitor the process stability every day, latent interactions lead to unwanted yield loss. To guarantee a high quality level, risky devices are additionally scrapped based on measurement data.

Within the framework of automation and industry 4.0, a major tool comprising of academic input across all technical disciplines is machine learning. While considered as a part of computer science, the field is substantially influenced by mathematics and statistics, as well as by engineering. Major applications reach from healthcare, business intelligence and natural sciences to those discussed in the scope of industry 4.0 - including automated decision support and decision making systems, robotics, etc. The idea of machine learning, which was mainly driven by the increase of computing power in the last decade, finally gained success and public reputation due to prestigious projects, such as *Watson*, developed by IBM [7] or Google's *Alpha Go* [8].

Inspired by the possibilities of data science, which is a rather new field of scientific research ranging between computer science, statistics and mathematics, as well as the demanding challenges raised in semiconductor industry, this thesis tackles the central problem of decision support for the purpose of process monitoring in wafer manufacturing. The goal is to develop a tool for detecting and recognizing process patterns in electrical measurement data, which indicate deviations in the production process. Machine learning and advanced statistical methods are essential parts to implement the central aspect of pattern recognition in such an intelligent system. In this respect, it is of major importance to choose adequate methods from the growing variety of algorithms in this field, to adapt them to fit the needs of the practical use-case and finally to combine them with knowledge available from human experts.

Appropriate machine learning algorithms are not directly capable of covering the required properties in certain practical cases - for instance, it must be specified how reliable training data can be gathered from the process, which kind of information is extracted to obtain the intended output or how the algorithm can detect and react on novel situations. Hence, the development of

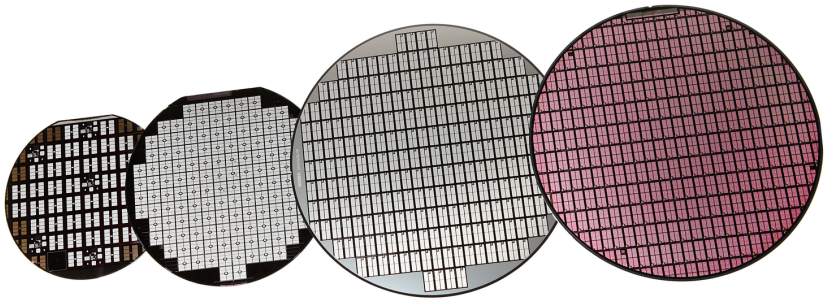


Fig. 1.2: A wafer is the processing unit, where up to several thousand devices are manufactured simultaneously during frontend production. Finally, the wafer undergoes the wafer test before proceeding to the backend processes. [10]

tailored methods is necessary - this introduces new challenges, which have to be resolved, leading to major scientific as well as industrial contributions presented in this work. From the industrial viewpoint, we propose an early warning system, which can be used to detect and recognize process deviations in measurement data, before they cause yield loss - hence, we pave the way for a more efficient production, going along with a sophisticated decision support tool for experts.

1.2 Background & SemI40 Project

The idea of deploying tools from the field of data science to semiconductor industry, especially to automate processes during fabrication, is the focus of the EU-project Power Semiconductor and Electronics Manufacturing 4.0 (SemI40) [9], initiated and coordinated by Infineon Technologies Austria AG. The project is intended to enhance the competitiveness of the European semiconductor industry and to respond to the needs associated with the transition towards industry 4.0. In the course of this project, 37 partners of 5 European countries contributed to 7 work packages and several distinct use-cases. This thesis is part of the use-case 1 in work package 3, entitled "Machine Learning and Automated Decision Making", which aims to extract and process information from datasets to enable decisions support and root-cause analysis. During the 3-year cooperation, major academic achievements for this project were made by KAI - Kompetenzzentrum Automobil- und Industrieelektronik GmbH in Villach, as well as Know-Center GmbH in Graz, while Infineon Austria AG contributed as a use-case provider and industrial supporter.

The concept of the work presented in this thesis involves the analysis of so-called *wafer test data*, i.e. data collected of electrical measurements after semiconductor frontend production to determine the functionality of manufactured devices. At this stage, up to several thousand devices are located on a slice of semiconductor material, the so-called *wafer*, see Fig. 1.2. After wafer test, devices passing the functional test undergo the backend processes, including sawing, bonding and packaging the devices. As wafer test is optimized towards indicating quality issues, i.e. violations of the specification negotiated with the customer, the usual procedure is to scrap malfunctioning devices. From the viewpoint of logistics, a bunch of usually 25 wafers is aggregated to a so-called *production lot* (lot) during manufacturing.

However, apart from the quality of a single device, even more information is contained in wafer test data: as devices are spatially distributed on the wafer and most production steps process the wafer as a whole, spatial dependencies are present (so-called process patterns). Such process patterns can be used for monitoring the production process, as they can indicate that process steps deviate from their proper behavior. For example, patterns can be provoked, if a layer is deposited in an inhomogeneous way. Detecting and correctly linking process patterns to their root-cause is a major objective for product experts, but highly prone to errors and subjective decisions if carried out in a manual way. Hence, providing automated tools for the analysis of wafer test data is rather a necessity than an option, due to the large amount of collected data and the sensibility of the results. In case of wrong decisions (e.g. fails of the product at customer level) severe economic consequences are possible. Fully relying on the judgment of human experts is not a feasible solution.

Existing tools are available in literature and can support the expert, but vary w.r.t. their targets. As an overview, most published research works on data analysis for wafer test data show the following characteristics:

- Many analysis tools for wafer test data focus on single-chip assessment (i.e. context of quality-related topics), but not on process patterns (i.e. context of process control).
- Out of those tools, which tackle pattern recognition, existing works describe methods for so-called pass/fail data, i.e. the results of comparing measurement data to the specification limits. Hence, production issues cannot be detected unless an error occurs and yield loss is triggered. Only few of these works can be directly transferred to processing the original measurement values (analog data).
- In general, if analog wafer test data are of interest, the focus is rather put on outlier detection than on pattern recognition in most cases. Moreover, pure pattern recognition systems cannot be considered as full decision support systems, as they neglect crucial information such as the expert knowledge or the degree of development of a pattern (which will be referred to as *intensity* in this work).
- Furthermore, most tools do not consider variations of process patterns, such as rotation or position change of a characteristic spot on the wafer.
- Classical data analysis tools usually do not consider the availability of limited labeled training data, as well as new pattern types depicting new kinds of deviations, which can result e.g. from new hardware configurations in production.

Further details on related work in literature is provided in Section 2.3, as well as in the method chapter, see Section 3.1, 3.2, 3.3 and 3.4.

1.3 Aim of the Thesis

Stated on a high level, the aim of this doctoral thesis is to present a valid, theoretically sound concept for an indicator that judges the presence and criticality level of process patterns in wafer test data by means of data-driven methods, combined with expert knowledge. Such an indicator will be called a Health Factor for Process Patterns (Health Factor) in this work. Concerning the development level, the goal of this thesis is to provide the theoretic framework for such an

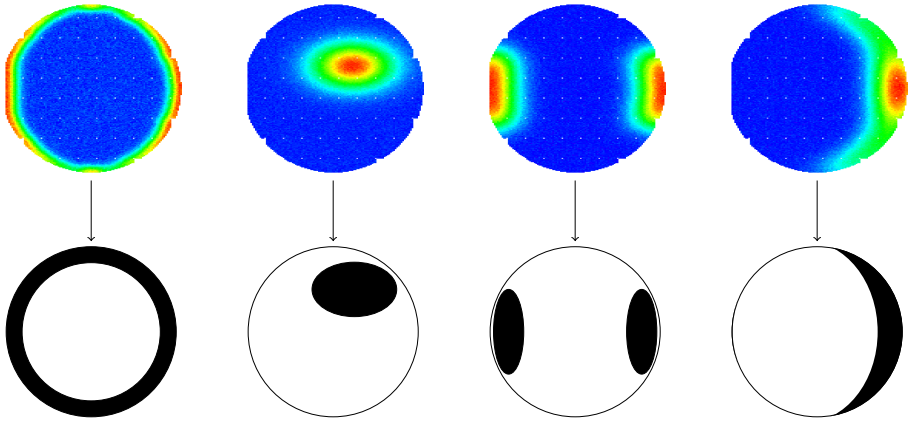


Fig. 1.3: The goal of RQ 1 is to extract informative image features (latent information) from wafer test data in order to distinguish between different pattern types.

indicator, as well as to verify it on exemplary datasets (proof of concept), finally obtaining a demonstrator for the suggested system.

While inspired by a practical goal, several scientific challenges have to be met along the way. We split these into 3 major research questions (RQs), given as follows:

RQ 1: How to automatically extract features for pattern recognition from (analog) wafer test data? (see Section 1.3.1)

RQ 2: How to recognize the pattern types available in the training dataset on new wafermaps? (see Section 1.3.2)

RQ 3: How to define a data-based "Health Factor for Process Patterns", describing the degree of affectedness of a wafer by critical process patterns? (see Section 1.3.3).

Although these RQs cover most of the scientific contribution of this thesis, further practical value is covered by the topic of the thesis, going beyond the scope of the RQs, e.g. when discussing data preparation and data structures in Section 3.1.

1.3.1 Research Question 1: How to automatically extract features for pattern recognition from (analog) wafer test data?

The first main challenge, which needs to be solved in order to define a valid Health Factor, is to solve the feature extraction problem for pattern recognition, i.e. to find a latent representation of the data, which covers the major characteristics of the pattern and neglects unnecessary side information. Due to the data structure and the characteristics of the problem, approaches from the field of image processing will be selected for this purpose. The addressed goal is to obtain a method, which is able to reduce data from a new analog wafer test dataset to the key information which covers present process pattern types. These pattern types are predefined by the product expert. The situation is illustrated in Fig. 1.3.

While feature extraction (with regard to distance-based dependencies on a spatial grid) is a typical and well-established problem in computer vision, typically tackled by extracting high-level features or applying a Convolutional Neural Network (CNN), this is a challenging task in the context of analog wafer test data. This fact is caused by the resolution (i.e. the number

of devices on each wafer), which differs between products, reaching from a hundred to tens of thousands of data points. Since wafer test data represent measurement values instead of image pixels, unbounded scales are another aspect to consider - in contrast to images, the scales of different measurement parameters can strongly differ from each other in addition, which makes comparisons difficult. Further, the large variety of process patterns (even including the possibility that new pattern types, i.e. classes occur), as well as by pattern-specific variation within each pattern class poses another problem.

For instance, pattern A could be defined by its specific position, e.g. consisting of a region at the border of the wafer. However, pattern B might occur at different positions on the wafer. As well, pattern C might change in size, while pattern D is specifically defined by its shape and constant spatial extent. All of these pattern-specific variations can be individually covered by image features, but in combination, distinguishing each pattern type from all others is difficult.

The procedure to tackle RQ 1 can be divided into 3 steps: preprocessing, feature extraction and definition of a distance measure. The latter represents the transition to the next step, i.e. the pattern recognition stage tackled by RQ 2. Preprocessing tackles the issues originating from data generation and format. These include measurement noise, patterns provoked by the test procedure, mixtures of patterns etc. Hence, preprocessing introduces methods to reduce noise, distinguish test and process influences and demix patterns. However, note that parts of the data preprocessing pipeline go beyond the scope of RQ 1, as the RQ primarily focusses on the extraction of relevant features for pattern recognition, while the preprocessing procedure is also an essential pre-step for each other approach tackled in RQ 2 and 3 (and therefore not a specific subtask in RQ 1). Details are provided in Section 3.1.

The main objective in RQ 1 is to extract the most informative features out of the wafer test dataset, in order to distinguish pattern types from each other. For this purpose, different feature types are taken under consideration, including classical, texture-based image features, such as e.g. Local Binary Pattern (LBP), as well as features delivered by a deep Convolutional Variational Autoencoder (CVAE). For details, see Section 3.2.

Finally, to observe whether the results obtained by feature extraction match the expectations of human experts, a suitable similarity or distance measure has to be used, measuring the degree of resemblance of two wafermaps. The definition of such a valid measure is closely related to the choice of the features. While the obtained features serve as input for the classifiers trained in RQ 2, the distance measure (as input for clustering algorithms) will be used for evaluation purpose only. Details on distance measures, together with suitable clustering methods, are provided in Section 3.3.1.

1.3.2 Research Question 2: How to recognize the pattern types available in the training dataset on new wafermaps?

Extending the concept obtained in RQ 1, RQ 2 tackles the classification task of analog wafer test data w.r.t. pattern types. After obtaining this information, the expert can provide further information on the pattern types, which is not intrinsically contained in the data, such as the information whether a pattern type depicts critical or uncritical process deviations. The pattern recognition task can be solved by machine learning tools, based on feature extraction in RQ 1. The goal is to categorize new analog wafer test data with respect to a set of pattern types

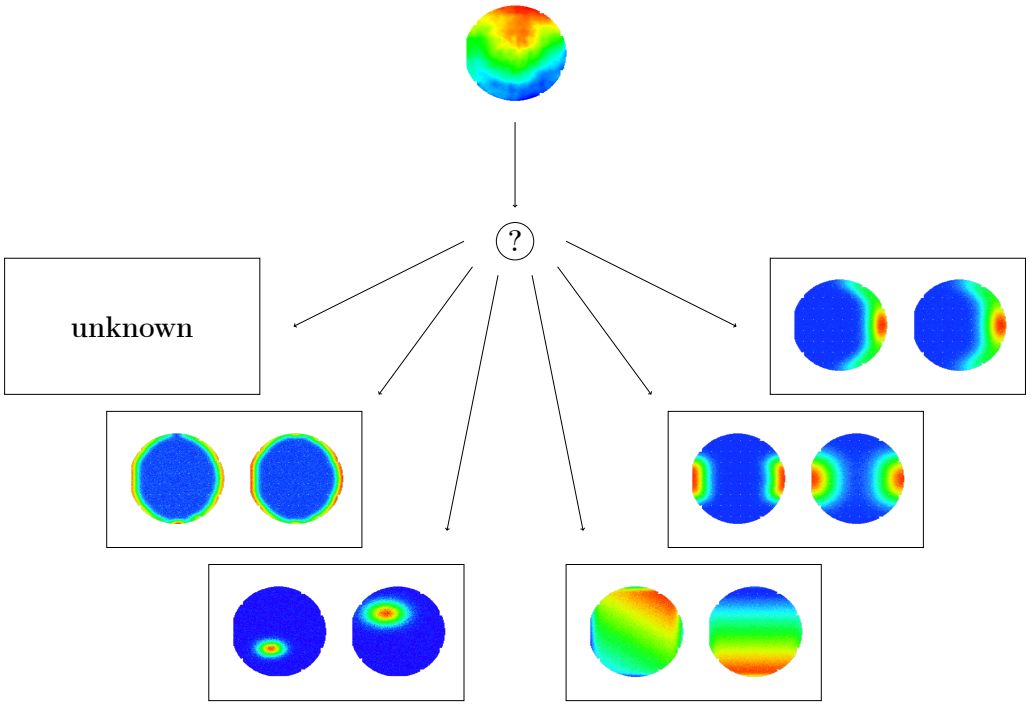


Fig. 1.4: The goal of RQ 2 is to classify wafer test data w.r.t. a set of predefined pattern types (i.e. classes), using the features extracted in RQ 1.

predefined by the expert. In order to avoid misunderstandings, this task will be denoted as *pattern type classification* in the course of this thesis. The concept is depicted in Fig. 1.4.

Basically, the idea of pattern recognition comprises multiple different problem definitions, including supervised classification tasks (usually with few classes of interest), as well as unsupervised and semi-supervised problems. Since in this thesis, specific patterns of interest (i.e. those, which indicate critical process deviations) need to be identified, purely unsupervised methods are hardly applicable. In order to introduce expert knowledge on the classes of interest, an expert provides a labeled training dataset showing prototypes of such classes. Thereby and by additionally assigning a *criticality level* to each class, the required extrinsic information is introduced. As an extension of the supervised pattern recognition approach, a semi-supervised method will be presented further.

Concerning the selection of a suitable method, a large variety of approaches is available in literature. This includes classical machine learning methods, such as logistic regression, decision trees (or random forests), Bayes classifiers, Support Vector Machines (SVMs), neural networks in various topologies, etc. Especially neural networks have gained popularity in the field of deep learning and therefore attracted attention in the scientific community. However, a major aspect when selecting a suitable method is the ability to handle a limited labeled training dataset, as well as interpretability and reproducibility. All of these criteria are hardly fulfilled when using neural networks, therefore more classical machine learning approaches will be preferred in this work. However, a detailed comparison of distinct algorithms, as well as a more detailed discussion on the topic of deep learning is provided in Section 3.3.2.

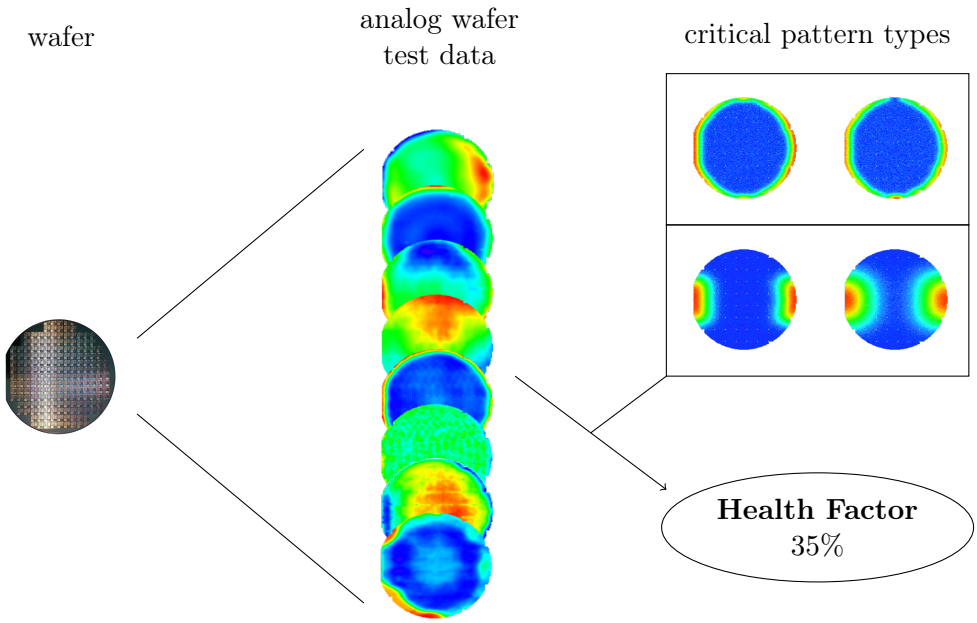


Fig. 1.5: The goal of RQ 3 is to define a Health Factor, i.e. an indicator for process deviations based on patterns occurring in analog wafer test data. The calculation is based on the recognition of critical process pattern types, tackled in RQ 2.

Although supervised learning achieves suitable results, one case cannot be handled: occasionally, if e.g. a new equipment is installed in production, a new type of pattern (i.e. a new class) can occur in the dataset. For this purpose, a more flexible method is needed, which is able to reject a new sample during evaluation, if it does not fit to any of the known classes. Such approaches (especially in a semi-supervised setting) are added to the approaches under consideration for this RQ. Details are provided in Section 3.3.3.

1.3.3 Research Question 3: How to define a data-based "Health Factor for Process Patterns", describing the degree of affectedness of a wafer by critical process patterns?

To achieve the final goal of the thesis, i.e. the definition of a Health Factor, which can be used to monitor analog wafer test data w.r.t. the presence and degree of development of critical process deviations, a combination of distinct components and information sources has to be implemented. First, the single components contributing to this final Health Factor must be identified. Secondly, they must be combined in a theoretically sound manner, fulfilling statistical and mathematical consistency properties. Finally, the Health Factor has to be provided in a scalable way, such that it can be evaluated on different aggregation levels (single electrical parameter, wafer, production lot, etc.) without losing its interpretability. The goal of RQ 3 is to aggregate all information collected before in one single numerical value (bounded to the $[0, 1]$ interval), indicating by 1 that a severe process pattern type is strongly forming on the wafer, while 0 indicates that hardly any process pattern is present. The schematic concept of the RQ is outlined in Fig. 1.5.

The key influencing factors for the Health Factor are dependent on the problem setup, where 3 major aspects are taken into consideration:

1. the *pattern type*, i.e. the type of pattern, which is present on the wafer,
2. the *pattern intensity*, i.e. the degree of development of the pattern on the wafer and
3. the *pattern criticality*, i.e. the severity level relating the process pattern to the root-cause in the production.

While components 1 and 2 are intrinsic properties of the data, the criticality level is provided as extrinsic information by the expert. Furthermore, the criticality level is related to a pattern type, not to a specific wafer or pattern instance.

Considering the pre-work done in the RQs 1 and 2, the components of pattern type and pattern criticality are tackled in more detail there. The pattern intensity quantification problem is an additional task, which is not explicitly covered within the 3 RQs, but will be presented in detail in Section 3.4.1.

The combination of the 3 Health Factor components is proposed by adapting the framework of statistical decision theory: the indicator is modeled as a loss (i.e. degree of severity), which occurs if a certain pattern type is detected for a critical pattern in combination with a high pattern intensity. In order to cover all possible situations, including the case that other than the presented algorithms are chosen as components, the Health Factor is defined in 3 versions: one for deterministic intensity and criticality, but probabilistic pattern type, another with probabilistic intensity and pattern type, as well as one with all components defined as probabilistic variables. Details are provided in Section 3.4

Finally, the Health Factor needs to be tracked at distinct aggregation levels. For example, an expert might intend to monitor the Health Factor at the level of production lots for products, which are known to be generally robust w.r.t. errors. Other products, where deviations occurred more frequently, might be investigated at a refined aggregation level to assure a detailed monitoring. This target will be achieved by defining the Health Factor at the lowest possible aggregation level and defining suitable link functions to elevate the level.

1.3.4 Disclaimer

Several aspects going beyond the scope of the presented RQs are not part of this thesis, although related to it. In particular, observing the link to the production process, as well as analyzing the root-cause of errors is left to the expert. These aspects would require further data sources (e.g. process control parameters from inline measurements), as well as specific process know-how. Similar topics are tackled in other research works, such as those resulting from the EU-project Integrated Development 4.0 (iDev40) [11] and require focussing on specific process blocks. However, in this work, we intend to provide a general framework, which is not restricted to specific parts of the wafer manufacturing process.

Another aspect, which is not in the scope of this thesis, is to present countermeasures or suggest concrete actions to the expert if a process deviation occurs. In this work, we propose an indicator for the existence of such deviations (i.e. a decision on whether an alarm should be triggered or not), whereas involving domain knowledge to provide more qualified decisions is not an objective. Such countermeasures would again require product- and process-specific input.

Furthermore, an economic evaluation of the impact of the suggested system on manufacturing is not in the scope of this thesis. Such evaluations would be specific to companies and products, which is not the aim of this work. Instead, solving the scientific and technical challenges to define a Health Factor based on methods from data science and statistics will be in the main focus.

Finally, the target of this research project is to present the theoretical basis, develop a suitable demonstrator and evaluate specific results as a proof of concept, whereas a final integration into semiconductor production will not be pursued. Such an integration would require resources on the company side, as well as specific evaluations beyond the general proof of concept presented here. In addition, such an integration project would require a strong industrial focus and individual solutions with low value from a scientific perspective. However, application-oriented projects to integrate data-driven methods into semiconductor production exist, see e.g. the EU-project Arrowhead Tools for Engineering of Digitalisation Solutions (Arrowhead Tools) [12].

1.4 Scientific Contribution

From the viewpoint of academia, this thesis tackles diverse RQs in the field of applied data science, as presented in Section 1.3. While these research topics are in accordance with the (practically inspired) goal defined above, the scientific aspects and novelties will be highlighted in this section. As an overview, Tab. 1.1 lists all publications, submitted to peer-reviewed international conferences and journals, which are related to the topics discussed in this thesis. In general, these works disseminate the main scientific findings constituting this thesis.

Contributing to RQ 1, the paper named "Markov Random Fields for Pattern Extraction in Analog Wafer Test Data" tackles the preprocessing stage, where image restoration techniques are tailored to use them for wafer test data. Moreover, the work on this topic is intensified in "An Explicit Solution for Image Restoration using Markov Random Fields", where the method is substantially optimized towards computational complexity. Furthermore, the focus on the feature extraction part of RQ 1 is extensively described in "Feature Extraction from Analog Wafermaps: A Comparison of Classical Image Processing and a Deep Generative Model", where a direct comparison between a classical feature extraction approach for images is compared to a CVAE on a wafer test data format.

For RQ 2, the paper named "A Comparison of Supervised Approaches for Process Pattern Recognition in Analog Semiconductor Wafer Test Data" focusses on supervised machine learning methods, applied on the classical feature set obtained from RQ 1. A major scientific novelty is invented in the paper "A Generative Semi-Supervised Classifier for Datasets with Unknown Classes", where a fundamentally novel concept of generalizing a semi-supervised Bayes classifier to detect previously unseen classes is described, providing a novel aspect in the field of machine learning. This work extends the objectives claimed in RQ 2 by the additional aspect of considering previously unseen classes.

Finally, "A Health Factor for Process Patterns - Enhancing Semiconductor Manufacturing by Pattern Recognition in Analog Wafermaps" illustrates and evaluates the complete concept of the Health Factor, describing the single components in detail. While summarizing specific parts of the previous works, the paper contributes to the field of decision support systems in cybernetics.

RQ	title	conference or journal	relation to this thesis	authorship and contribution
1	Markov Random Fields for Pattern Extraction in Analog Wafer Test Data [13]	International Conference on Image Processing Theory, Tools and Applications (2017)	Section 3.1	first author, theory and experiments
	An Explicit Solution for Image Restoration using Markov Random Fields [14]	Journal of Signal Processing Systems (2019)	Section 3.1	co-first author, theory and related work
	Feature Extraction from Analog Wafermaps: A Comparison of Classical Image Processing and a Deep Generative Model [15]	Transactions on Semiconductor Manufacturing (2019)	Sections 3.2 and 3.3.1	co-first author, theory and experiments for approach (A)
2	A Comparison of Supervised Approaches for Process Pattern Recognition in Analog Semiconductor Wafer Test Data [16]	International Conference on Machine Learning and Applications (2018)	Section 3.3.2	first author, theory and experiments
	A Generative Semi-Supervised Classifier for Datasets with Unknown Classes* [17]		Section 3.3.3	first author, theory and experiments
3	A Health Factor for Process Patterns - Enhancing Semiconductor Manufacturing by Pattern Recognition in Analog Wafermaps [18]	International Conference on Systems, Man and Cybernetics (2019)	Section 3.4	first author, theory and paper concept
	Process Monitoring in Industry 4.0 - A Framework for Detecting Process Deviations based on Pattern Recognition* [19]	Transactions on Industrial Informatics (2019)	Section 3.4	first author, theory and paper concept

Tab. 1.1: List of scientific publications related to this doctoral thesis. * indicates that manuscripts were not yet accepted for publication at the time of thesis submission.

RQ	title	conference	relation to this thesis	authorship and contribution
0	Health Factor for Process Patterns	SamI40 Workshop at iKNOW (2016)	Section 1.2	first author, theory and paper concept
	Data Science along the Semiconductor Frontend Production	Statistische Woche (2018)	Section 2.3	second author, ideas for pattern recognition
1	An Information-Theoretic Measure for Pattern Similarity in Analog Wafermaps [20]	European Advanced Process Control and Manufacturing Conference (2019)	Section 3.3.1	co-first author, experiments
2	Machine Learning Techniques for Automated Wafer Health Assessment using Wafer Test Data	European Advanced Process Control and Manufacturing Conference (2019)	Section 3.3	second author, ideas for machine learning methods
3	An Investigation of Statistical Measures for Intensity Comparison of Process Patterns in Analog Wafer Test Data [21]	European Advanced Process Control and Manufacturing Conference (2019)	Section 3.4.1	second author, ideas for statistical tools

Tab. 1.2: List of contributions at industrial and informal scientific venues related to this doctoral thesis. Publications assigned to RQ 0 describe the general concept of the work without specifically tackling any of the RQs.

Specific parts of the scientific contribution presented in this thesis were developed as sub-projects in a separate way, e.g. as a continuation of previous works or in accompanying Master's theses by students supervised in the course of this project. In particular, these topics include the methods for intensity quantification of process patterns in analog wafer test data, Section 3.4.1, as well as the methods for blind source separation, test pattern removal and the optimization of the Markov Random Field approach for preprocessing analog wafermaps in Section 3.1. Furthermore, the Deep Learning approach evaluated for Feature Extraction in Section 3.2.4 was developed and applied by research partners at Know-Center in the course of the collaboration.

1.5 Industrial Contribution

In order to cover both, the academic and the industrial challenges, further dissemination was done at application-oriented venues. Tab. 1.2 contains information on such contributions.

Concerning the relation to this thesis, both "Health Factor for Process Patterns" as well as "Data Science along the Semiconductor Frontend Production" provide a general overview and outline on the topics discussed in the course of this thesis. While the first presents the specific idea of constructing a Health Factor, the latter focusses on the abilities related to data science in production environments. Although they might not have a high visibility from a scientific

perspective, such publications demonstrate the need and the possibilities of data-driven methods in industry and are therefore essential to guarantee support from the industrial side.

"An Information-Theoretic Measure for Pattern Similarity in Analog Wafermaps" presents an alternative distance measure for features extracted from wafermaps. It is evaluated by clustering methods, yielding accurate results.

The paper named "Machine Learning Techniques for Automated Wafer Health Assessment using Wafer Test Data" focusses on the abilities related to machine learning methods for industry. In specific, the subtasks of this project, which can be tackled with the methods presented in Section 3.3 are depicted.

Finally, "An Investigation of Statistical Measures for Intensity Comparison of Process Patterns in Analog Wafer Test Data" summarizes the approaches to describe an intensity measure for the development of process patterns on wafermaps. This topic is related to the last RQ, where pattern type and criticality are combined with such an intensity measure to obtain a final Health Factor.

In summary, the main industrial contributions are to the following:

- **prevention of yield loss**: due to the usage of analog instead of pass/fail wafer test data, errors can be detected before they lead to yield loss.
- **decision-support for experts**: the manual effort of experts is reduced by providing a system to automatically screen a wafer test dataset for critical patterns.
- **transferability to different products**: the presented system is mostly independent of specific product characteristics, but can be transfered to any other product or technology.
- **integration of machine learning in manufacturing**: the Health Factor presented in this work allows to implement machine learning tools into semiconductor production to benefit from the capabilities of data-driven systems.

Hence, this thesis is intended to contribute to both, scientific and industrial goals. Both aspects will be highlighted in the course of this work.

1.6 Structure of the Thesis

Following this introduction, Chapter 2 discusses the tackled problem and its background in a more detailed way. This includes an overview on the semiconductor production process in whole, provided in Section 2.1. The focus on the wafer test stage, as well as the according dataset structure is emphasized in Section 2.2. While the main intention behind wafer testing is to assess product quality, the topic of pattern recognition is investigated in more detail in Section 2.3. Finally, in this chapter, we introduce the formalized problem definition based on the preceding information provided.

While the main focus of Chapter 2 is put on semiconductor production, a set of methods is presented in Chapter 3, tackling the single RQs and their subproblems individually. Section 3.1 focusses on the selection of the most suitable data format for the subsequent calculations, followed by Section 3.1, which tackles data preprocessing in detail. This includes several aspects, i.a. normalization, outlier removal, missing value imputation, blind source separation and image

restoration. The pattern recognition problem is investigated in Section 3.2, focussing on feature extraction, followed by suitable machine learning methods in Section 3.3. Section 3.4 combines all previously introduced methods to define a mathematically sound Health Factor, fulfilling the requirements proposed in Section 2.4.

To bridge the gap between theory and application, we apply the suggested methods to simulated, as well as real-world datasets in Chapter 4. For this purpose, the datasets and their characteristics are presented in Section 4.2. The experiments are divided into three parts: first, in Section 4.3 the performance of the pattern type recognition methods is evaluated, while Section 4.4 contains the results for the intensity quantification problem. The full Health Factor concept undergoes a detailed investigation in Section 4.5. While major benefits and drawbacks are stated within these segments, details and limitations going beyond these aspects are discussed in Section 4.6.

This work is concluded with Chapter 5, where we summarize the main results in a short resume, provided in Section 5.1. The main findings, as well as lessons learned are stated in Section 5.2. Future work, including both, open academic and industrial aspects, is announced in the outlook, see Section 5.4.

Due to the fact that this thesis originates from a research project with multiple collaborators, the authors will be referred to as "we" for reasons of consistency. However, all parts except for those explicitly stated otherwise are attributed to my work.

2 Pattern Recognition in Semiconductor Manufacturing

2.1 Semiconductor Manufacturing

Since this work focusses on the development of appropriate data science methods for application in semiconductor industry, an overview of semiconductor manufacturing will be provided first. In particular, specific challenges and characteristics of the products, as well as major process steps will be explained in this section.

2.1.1 Products and Related Challenges

As a result of the increasing demand, semiconductor companies are forced to scale up their capacities. Reducing the product size while increasing the wafer diameter from 200 to 300 mm for many products are characteristics of this ongoing process. However, shrinking the device causes a problem, in particular for power semiconductors: due to the physical fact that a smaller area implies a higher resistance, this results in a higher loss of energy by device heat-up. Consequently, semiconductor devices are required to be as thin as possible to counteract this effect, reaching a range of few micrometers. With this development, manufacturing complexity has grown excessively: handling larger and thinner wafers, each carrying thousands of tiny devices permits no error during production. Modern 300 mm thin-wafer technologies, first accomplished by Infineon Technologies as announced in 2011 [22], have a pioneering role in this respect.

Another consequence resulting from the large economic interest is the competition for more powerful product specifications in order to dominate the market. New materials promise to meet such expectations and outperform silicon as a classical semiconductor element, including Silicon Carbide (SiC) and Gallium Nitride (GaN), see [23] and [24], respectively. However, new challenges emerge from the fact that the properties of these materials are not sufficiently researched, yet. Hence, new types of errors occur, increasing the need for extensive control of each manufacturing step to detect deviations in each stage. In detail, main technological goals of the semiconductor industry are set and formulated in the International Technology Roadmap for Semiconductors (ITRS) [25] and International Roadmap for Devices and Systems (IRDS) [26] documents.

The product palette of semiconductors is as large as their fields of application - reaching from fundamental tasks, such as switching an LED diode, up to microcontrollers performing complex tasks by integrated software components.

Fields of applications of semiconductor products cover classical areas, such as memory chips, power switches or diodes, as well as newer areas, such as microprocessors, photovoltaic or smart

sensors. Integrated circuits (ICs) are designed to intrinsically perform complex tasks. Depending on the specific use, such microcontrollers have an astonishing performance, which can be exploited by optimized software solutions. They handle complex general-purpose tasks on smartphones, as well as they process high-performance cryptographic functions in sensitive security applications. In contrast, so-called Integrated-Gate Bipolar Transistor (IGBT) devices (e.g. used in trains or windmills) are power devices, tailored to switch very high loads. Sensors are another growing branch of semiconductor industry, which is especially pushed by robotics and autonomous driving, where tons of sensors are required. These include optical sensors, as well as pressure sensors (e.g. tire pressure), acoustics, etc. Photovoltaic cells combine the need for optical sensors with the requirement of switching high power loads.

2.1.2 Overview on the Production Process

Due to the broad range of semiconductor devices existing across all areas of electronics, the detailed production process steps vary from product to product. In general, the manufacturing process takes several weeks to months, depending on the product complexity. A fundamental workflow is common to all products and technologies, starting from the pure silicon slice (bare wafer) and ending with the final product (device), which is sold to the customer, see [27]:

1. Frontend production
2. Wafer Test
3. Backend production
4. Device (or IC) Test

An overview on this process is provided in Fig. 2.1, where the single stages are illustrated. Detailed explanation on the stages is provided e.g. by Klemmt [28].

Frontend production Frontend production covers the process steps, which are performed on the whole wafer. This stage is divided into Front End of Line (FEOL) and Back End of Line (BEOL). At the beginning of FEOL, the wafer slice is cut out of monocristalline semiconductor material. After cleaning the wafer surface, trenches are etched into the substrate (i.e. the ground material). As a result, rough structures are formed on the wafer, including a gate module and a source and drain module. Implantation is a further process step, where ions are fired into the material in order to reach electrical properties (doping), followed by a heat-up called Rapid Thermal Processing (RTP), which provokes a diffusion in the material. A sequence of metal layers is applied during BEOL and connected using inter-metal dielectrics. Finally, the wafer undergoes a passivation step to protect it from environmental influences.

Major frontend production steps include the following, recombined in a product-specific order:

- lithography
- implantation
- etching
- RTP
- deposition

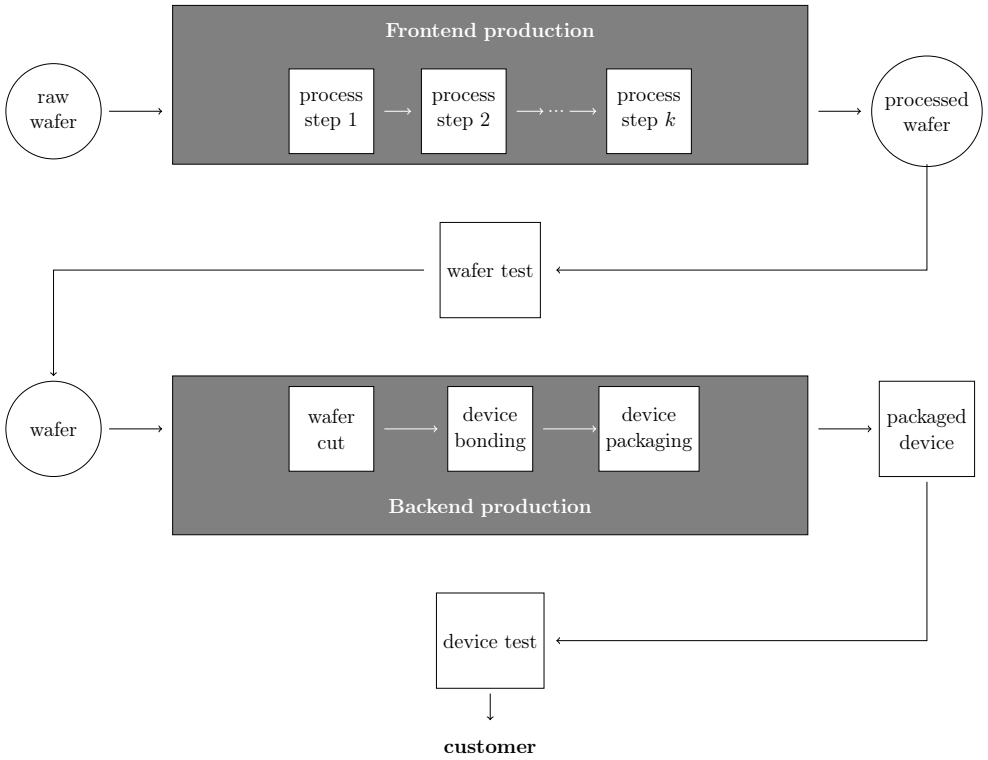


Fig. 2.1: An illustration of the semiconductor manufacturing process, divided into the central stages of frontend production, wafer test, backend production and device test.

- polishing and cleaning

Many process steps are iterated multiple times for a single wafer. For example, etching is performed for several distinct layers, where different methods (e.g. chemical etching, plasma etching, etc.) are deployed. In general, frontend production is a technical as well as logistical challenge in order to guarantee the correct order of process steps and, at the same time, fulfill time and capacity constraints in the production line.

In course of the frontend production, so-called Advanced Process Control (APC) and inline data are collected, consisting of equipment parameters used for monitoring the process as well as of inline measurements, including e.g. optical inspections for particles or layer thicknesses. Since on the one hand, equipment data are not directly linked to the product but rather to the observed hardware and on the other hand, inline measurements do not cover all single devices on the wafer, there is no exhaustive functional test until the wafer test is performed after frontend production.

Wafer Test In order to assess the quality and performance of products on wafer level, wafer test is performed after frontend production. Wafer test includes a sequence of electrical parameters measured on each device. Each device is contacted by a set of needles on a probecard, while the wafer movement is controlled by the so-called wafer prober. The whole system of wafer prober and wafer test platform is called Automatic Test Equipment (ATE), see Fig. 2.2a. As a result, resistances, voltages and currents can be measured between drain, gate and source (in case of a simple switch). Furthermore leakage currents, i.e. currents occurring although the device is switched off, are of major interest. For more complex products, e.g. microcontrollers, a broad

range of functions must be covered by the wafer test. Hence, the complexity of the test procedure depends on the products and can consist of up to thousands of test parameters.

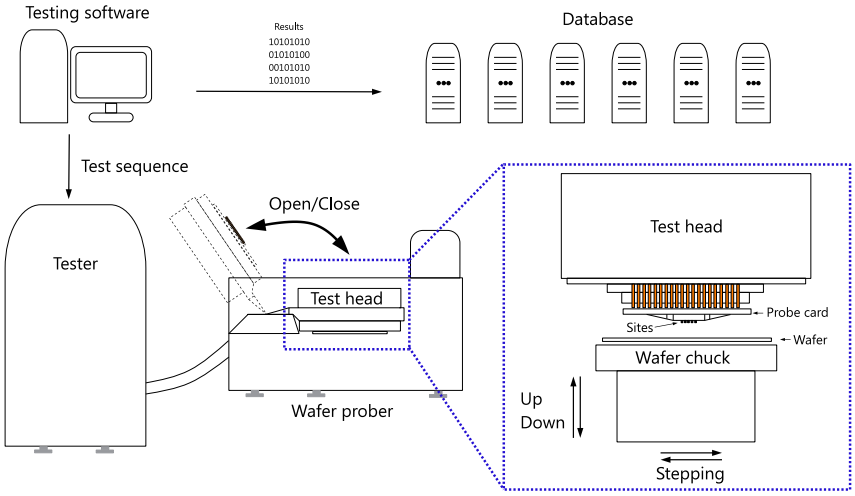
As contacting each device on the wafer sequentially is time-consuming, performing parallel tests has shown to be the most effective testing procedure, see Rivoir [30]: for this purpose, the test head contains a probe card with several needles, such that multiple devices can be measured in one single step (so-called touchdown). The single positions of the probe card are referred to as sites. Typical examples for needle arrangements are presented in Fig. 2.2b. Measuring only one device per touchdown is rather used for large products (i.e. few devices per wafer), while small products are measured in a highly parallelized way. According to a predefined scheme, the test head steps over the wafer (or, in detail, the wafer is moved to the test head on a chuck). This stepping procedure is optimized w.r.t. testing time.

Depending on the magnitude of the test program, wafer test can take several hours for one production lot. After testing, the measured test values are compared with upper and lower thresholds for each parameter (specification limits), indicating whether the performance of the device fulfills the requirements. If a violation occurs in a single parameter, the device is scrapped before further processing. Most test program settings prescribe that no further tests are conducted on such a device, resulting in missing values on the wafermap for subsequent parameters. The resulting dataset, containing measurement values as well as meta data (such as specification limits, x-y positions, etc.) are stored in databases as so-called wafer test data, discussed in Section 2.2 in more detail.

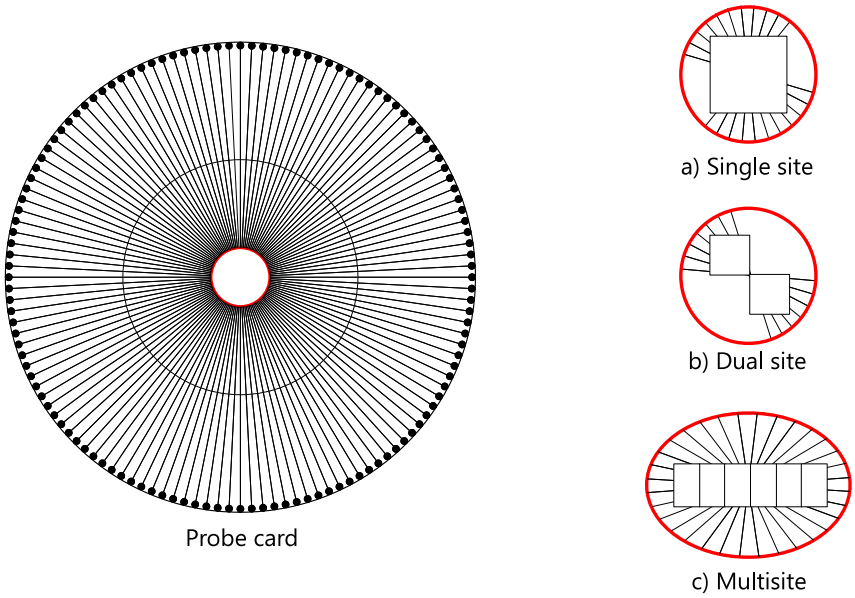
Backend production In exceptional cases, semiconductor devices are directly sold after wafer test, so-called bare die products. In all other cases, backend processes are applied. As a first step, the wafer is cut into the single devices, each fabricated individually from this time point onwards. Usually, single-chip tracking is not possible in many subsequent steps, i.e. all devices are merely identified by their wafer and lot number, but their x-y positions on the wafer do not persist. Hence, single-device anomalies occurring in backend cannot be traced back to parameters measured during wafer test for the specific device.

After cutting the wafer, the backend processes mainly consist of bonding the devices (i.e. fixing copper wires to apply electrical loads) and packaging them via a mold compound to protect from external influences. While frontend processes are mainly responsible for preparing the functionality of a device, backend rather ensures protection and robustness.

Device Test At the end of the production pipeline, final tests are performed to assure the product quality. Apart from functional tests, e.g. detecting whether all wire bonds are fixed correctly, these include Burn-In (BI) tests, where thermal cycling is applied to reduce the risk of infant mortality at the customer. In general, electronic devices are known to show higher failure rates directly after production (infant mortality phase), as well as after exceeding the standard life period (wear out phase). In between these phases, the failure rate is low, resulting in a reliable product. Hence, BI is aimed at overcoming the infant mortality phase, directly entering the phase of low failure rates when sold to the customer [31, 32]. However, BI testing is expensive w.r.t. time and test resources. Recent research projects therefore intend to reduce



(a) Wafer prober.



(b) Chuck for parallel measurements.

Fig. 2.2: The test equipment used to conduct wafer test for semiconductor devices (wafer prober or ATE) is depicted in 2.2a in a schematic way. In order to conduct tests on multiple devices in parallel, different probe cards are in use, see 2.2b. [29]

Lot	Wafer	X	Y	SITE_NO	$Test_1$	$Test_2$	...	$Test_n$	HBIN	SBIN	P/F
lower specification limits					$1.25e^{-3}$	$3.49e^{-5}$	...	0			
upper specification limits					$2.64e^{-2}$	$8.51e^{-5}$	...	$2e^{-2}$			
Lot1	1	10	15	1	$4.26e^{-3}$	$8.37e^{-5}$	...	$1.02e^{-2}$	0	0	P
Lot1	1	10	16	2	$5.02e^{-3}$	$4.53e^{-5}$	...	$9.85e^{-3}$	0	0	P
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮	⋮	⋮	⋮
Lot1	2	10	15	1	$2.32e^{-3}$	$\mathbf{8.61e^{-5}}$	...	•	2	4	F
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮	⋮	⋮	⋮
metadata					test parameter values				test results		

Tab. 2.1: Data structure of wafer test data. • marks that a parameter is not observed as the device is already classified as "fail" according to the preceding test parameters. A test value exceeding the specification limits is marked (bold).

the effort spent for BI testing by providing better statistical predictions on risky devices based on other data sources. One example using wafer test data is provided by Zernig et al. [33].

After passing the device test, semiconductor products are sold to the customer, where they are integrated into electronic systems. Due to the high quality standards in many industries, the overall yield, i.e. the percentage of devices passing all test stages, can be quite reduced due to process variations, causing high production costs for manufacturers [34]. Therefore, the yield is an essential key number to evaluate the business success in semiconductor manufacturing. Usually, yield is subdivided in frontend yield (percentage of devices passing wafer test) and backend yield (percentage of devices passing device test).

In this work, main focus will be put on frontend processes, where wafer test is considered the final stage for these investigations. Hence, yield will refer to the frontend yield, neglecting devices that fail during backend and device test.

2.2 Wafer Test Data

As wafer test data are in the main focus of this thesis, more details with regard to their structure and format will be provided here. A large variety of information can be extracted from this dataset - going beyond the comparison of wafer test parameters to specification limits, more information can be retrieved out of this dataset.

2.2.1 Characteristics of Wafer Test Data

The data structure of a typical wafer test dataset is presented in Tab. 2.1. In general, wafer test data consist of 3 segments: metadata, test parameter values and test results.

Metadata consist of logistical information, such as the lot and wafer number, as well as the coordinate of the device under test and the site of the probe card, which was used for testing. Further columns can include the revision of the test program, the measurement date and time, etc. In general, metadata are essential as background knowledge in order to interpret the results obtained from the test parameter columns.

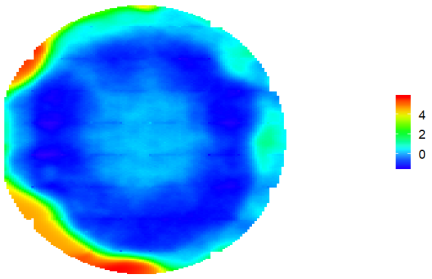


Fig. 2.3: A wafermap, depicting the values of an electrical parameter during wafer test. According to the color scale provided in the legend, red indicates high values, while blue corresponds to low values.

The second category consists of the values obtained from the test procedure: the test parameters. Here, each observed electrical test parameter ($\text{Test}_1, \ldots, \text{Test}_n$) corresponds to one column in the dataset. Note that these test columns are associated with different physical units (e.g. voltages, or resistances), hence the data scales can vary from column to column. For each of these test parameters, upper and lower specification limits are additionally provided, defining the range, which must be reached by the device in order to pass the wafer test. In case that a device exceeds any of these thresholds, the device is automatically scrapped and the subsequent tests are not conducted, resulting in missing values in the dataset (marked by •). Such original test parameter values are commonly denoted as *analog wafer test data* or continuous wafer test data.

Finally, the wafer test dataset contains test results and aggregated key values obtained from the test parameters. These include Hard Bin (HBIN) and Soft Bin (SBIN) entries, i.e. categories to locate the test parameter of the device, which exceeded the specification limits (0 indicates that all specification limits were fulfilled). These values provide the expert with information on the type of error. Furthermore the pass/fail (P/F) column summarizes whether the device passed the wafer test or not.

2.2.2 Wafermaps

A typical illustration of wafer test data is obtained via so-called *wafermaps*: the idea is closely related to plotting heat maps, i.e. the test parameter values obtained for each single device on the wafer are represented by coloring the according x-y position on a spatial grid. Hence, the color scale corresponds to the value scale of the electrical parameter - in this work, the color scale will be chosen such that red indicates high and blue indicates low values. In general, a wafermap is obtained for each measured test parameter in the test procedure and for each wafer. Wafermaps clearly illustrate the spatial dependencies of neighboring devices on the wafer, but due to the high number of wafermaps resulting from a wafer test dataset, detailed manual analysis is impossible. For instance, if a production lot contains 25 wafers and the wafer test procedure covers 100 test parameters, a number of 2500 wafermaps will be obtained. An exemplary wafermap is depicted in Fig. 2.3.

Wafermaps are closely related to image data, showing similarities such as a regular x-y grid, colorings, etc. However, five major differences have to be outlined:

- **Round border**
  While images are, by definition, rectangular, the values shown on wafermaps follow the structure of the wafer. As wafers have a circular shape, the outer border of the wafermap is nearly circular (for quadratic device types) or elliptic (for rectangular device types).
- **Missing values**
  Structural differences exist also within boundaries of the wafer: missing values can occur due to two reasons: firstly, a violation of the specification limits triggers a stop of the test procedure for a device, i.e. the device is marked as missing value in all subsequent wafermaps. Secondly, specific positions on the wafer are reserved for Process Control Monitoring (PCM) structures, used for additional measurements after frontend production. No device is manufactured at these positions and hence, no test can be performed there.
- **Single channel**
  A wafermap corresponds to a single-channel (greyscale) image, although the colors might (optically) suggest an RGB-type image. The coloring is rather used for depicting value differences more clearly and explicitly marked in the image legend. Nevertheless, a wafermap is basically of a singe-channel type and must not be interpreted as an RGB image.
- **Unbounded scales**
  In contrast to images, where the pixel values are bounded by $[0, 255]$ on an integer scale, wafermaps depict test parameters, which can be considered as unbounded. Hence, a normalization of the data range is a crucial preprocessing step - for this purpose, the specification limits can be of great value.
- **Low resolution**
  Finally, a wafer contains hundreds or several thousands of devices, corresponding to an image of the same number of pixels. In comparison with high definition images used in many applications of computer vision, covering millions of pixels, the resolution is in a very low range and changes from product to product.

While in principle, methods from the field of image processing are applicable on wafermaps, these aspects have to be taken into consideration.

2.2.3 Analog vs. Pass/Fail Data

Frequently in semiconductor industry, the term "wafer test data" is associated with the Pass/-Fail column in the dataset of Tab. 2.1. The same is true for wafermaps (so-called *pass/fail wafermaps*), where red indicates fail devices, while green is assigned to devices passing wafer test. A more detailed overview is provided if the devices in wafer test are assigned to different colors according to their HBIN or SBIN values. Using such a *bin wafermap*, the expert is able to distinguish between different error mechanisms coinciding on a wafer. Another possibility to consolidate information from wafer test in wafermaps is to use *stacked wafermaps*, i.e. a map depicting the percentage of devices passing wafer test evaluated for each single x-y coordinate throughout a lot.

By default, we will not refer to any of the variants presented above in this thesis. Instead, analog wafer test data and analog wafermaps, depicting single test parameter values, will be used unless explicitly stated otherwise. The advantage of analog data is that they cover more information

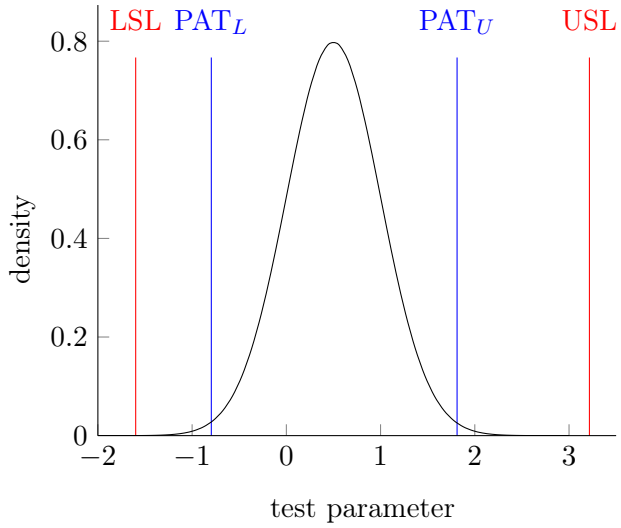


Fig. 2.4: Lower and upper specification limits, LSL and USL (red), as well as lower and upper Part Average Testing (PAT) limits, PAT$_L$ and PAT$_U$ (blue) are applied to the test parameter values resulting from wafer test. PAT detects outliers, which are not removed by the specification limits.

than other types of wafer test data, which are aggregated. Hence, analysis of analog data can reveal latent information, which is available before an error occurs.

The disadvantage of analog wafer test data is their large volume: up to thousands of wafermaps are required to cover the information obtained from wafer test for one single lot, involving all distinct test parameters. In general, correlations between test parameters, which are commonly present in wafer test data, are not covered when observing analog wafermaps. To handle the large amount of analog wafer test data and the broad spectrum of information covered therein, we propose automated tools for taking decisions based on data-driven algorithms.

2.2.4 Statistical Tools for Wafer Test Data Analysis

In case that the yield of a wafer is low, i.e. a significant percentage of devices on the wafer fails, industrial standards additionally require further actions, see e.g. the AEC - Q002 standard for statistical yield analysis in the automotive sector [35]. In case of an exceptionally high number of failures, a wafer or even a whole production lot is scrapped.

In order to detect latent effects, which cannot be detected using the specification limits presented in Section 2.2.1, statistical measures are additionally applied on the analog wafer test datasets. These statistical tools include outlier screening methods, e.g. PAT, as prescribed by the quality standards for automotive industry, AEC - Q100 [36]. The underlying justification for these methods is the assumption that test parameter values deviating from the main distribution of the respective test (outliers) contain a high risk of failing early in application [33]. Hence, so-called PAT-limits are introduced in addition to the specification limits, but represent statistical risk instead of physical failure. Fig. 2.4 illustrates how specification and PAT limits are applied to scrap devices violating these limits, compared to the histogram of the parameter.

However, PAT is merely suitable for one-dimensional, Gaussian distributed data, but can be extended by more powerful statistical tools (e.g. multivariate outlier screening techniques).

Examples for such multivariate outlier detection methods used in semiconductor industry are multivariate PAT [37], as well as Nearest Neighbor Residuals (NNR) PAT, described in [38] and subsequent works. These methods are summarized in [33].

For multivariate PAT, a multivariate Gaussian distribution is assumed to cover outliers emerging in mutually correlated test parameters within the test program. Hence, several data points are available for each single devices. In this case, outliers can take values, which are within the distribution of each single component, although deviating from the multivariate distribution. Hence, it is not sufficient to apply an outlier method on single components. Instead, the parameters of a multivariate Gaussian distribution must be estimated.

While multivariate PAT enables to take correlations between distinct tests into account, NNR PAT generalizes the PAT method w.r.t. spatial correlations among devices on the wafer. The preceding methods can merely detect outliers if they show differences compared to the global data distribution, but not necessarily if they deviate from their local neighborhood. Therefore, NNR is applied as a transformation before applying PAT in order to robustly eliminate the spatial trend on the surface of the wafer. A similar approach is taken in a one-dimensional case for time series analysis, where the trend parameter can be removed by calculating pairwise differences between values at subsequent time points, e.g. described by Alagić [29]. NNR transforms the test parameter values of each point by subtracting the median over its spatial neighborhood.

Other statistical methods for detecting outliers and single-device abnormalities in the measurement data are available, but industrial standards mainly focus on the basic approaches mentioned above. However, evaluating NNR reveals that most wafermaps are affected by spatial trends, containing regions deviating from the main data distribution, as well as gradients or regularities (e.g. rectangular structures) on the wafer. Other than outlier detection methods investigated in the research works mentioned above, detection and assessment of these patterns will be the main focus of the present work.

2.2.5 Process and Test Patterns

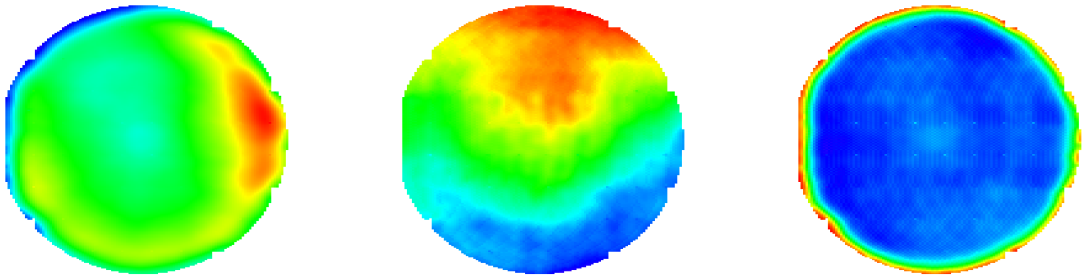
As described in the previous sections, regularities are commonly present on wafermaps, suggesting that an assumption of independent and identically distributed (iid) data will not hold for the devices on a wafer. Instead, especially spatial dependencies, i.e. covariances that depend on the distance between the observed points on the wafer, have to be considered.

Technically, such patterns are of various sizes, shapes and characteristics, but can be traced back to one of the following two root-causes: process deviations (process patterns) and testing errors (test patterns).

Process Patterns

Process patterns originate from deviations, errors or other influences during the frontend production process. Such anomalies are unintended as, by definition, each device should be manufactured in exactly the same way. However, this assumption does not hold in practice. Instead, production processes tend to show an inhomogeneous behavior over the wafer.

For example, many material layers are applied in a circular way from the middle to the border of the wafer. Slight variations in these processes can provoke that border regions show a slightly thicker or thinner layer thickness than the center regions, resulting in different higher (or lower) wafer test parameter values in the border regions (border patterns). Another example is that a



(a) Process pattern 1 (b) Process pattern 2 (c) Process pattern 3

Fig. 2.5: Wafermaps showing different process patterns. Usually, the error-sources of such patterns can be traced back to the production process by experts.

(a) Site pattern (b) Touchdown pattern

Fig. 2.6: Wafermaps showing different test patterns.

carrier of the process equipment leaves an imprint at a specific position on the wafer, resulting in a cluster of outliers visible in the wafer test data. Further, even more severe errors, such as wrong machine parameters leading to longer processing times than scheduled can be identified by specific process patterns, see e.g. Fig. 2.5.

Test Patterns

In contrast to process patterns, which originate from the frontend process steps, test patterns are provoked by the wafer test procedure, see e.g. Fig. 2.6. Mainly two types of test patterns are possible:

Firstly, a single needle on the probe card might be deformed or contaminated by particles, resulting in wrong test parameter values at the affected site. Such *site patterns*, as shown in Fig. 2.6a, are characterized by their regularity, provoked by the stepping procedure (see Section 2.2). If, for instance, one site is damaged in an 8-fold multisite measurement, each 8th device w.r.t. the stepping procedure shows a clear offset compared to all other values, i.e. is an outlier. When a test pattern is detected during wafer test procedure, maintenance or an exchange of the probe card is triggered.

Secondly, a single touchdown, i.e. the procedure of moving the wafer chuck towards the probe card, can fail. In this case, all sites are either too far away from the wafer or too close, such that the measurement values at all sites in this touchdown are distorted. Optically, a touch-

down pattern consists of a sharp-edged region, deviating in the measurement values from their neighborhood. An example for a *touchdown pattern* is provided in Fig. 2.6b.

While process patterns are of high interest when observing the production process or the product quality, test patterns interfere such evaluations. Indeed, test patterns contain wrong measurements, which might result in wrong decisions when being confused with process patterns. However, due to limited resources, outlier devices are rather scrapped than retested when wafers or lots are affected by test patterns.

Usually, both types of test patterns can be distinguished from process patterns by eye. However, for automated procedures, this distinction is challenging. In Section 3.1.4, we provide methods to detect and remove test patterns as preprocessing for the subsequent analysis of process patterns.

2.3 Process Pattern Recognition

As presented in Section 2.2.5, process patterns are provoked by deviations during frontend production and, hence, experts observed that they contain information on such anomalies. This information can be exploited by automated tools to enhance process monitoring and early error detection. Currently, the analysis of process patterns is a major topic of interest in the research community on semiconductor manufacturing, although existing research works aim at investigating pass/fail or bin wafermaps instead of analog wafer test data.

2.3.1 Requirements for Process Pattern Recognition

In general, two main objectives are pursued by analyzing wafer test data w.r.t. process patterns: gaining information about the frontend production process (process view, depicting what has happened before the test, e.g. process deviations) or collecting information about the quality of the produced devices (product view, depicting what will happen after the test, e.g. early device failure). While the latter is closely related to the statistical outlier detection methods described in Section 2.2.4, the main focus of this thesis is put on the gaining information on process deviations. However, the provided methods can also be applied for other purposes.

Particularly, the assumption taken in the following states that process deviations provoke specific process patterns in the wafer test dataset. Such process patterns can be of various shapes, sizes and intensities, but aggregate to certain *pattern types*, i.e. different variations of patterns provoked by the same root-cause.

Assumption 1 (Identifiability of Process Deviations)**.** *Each process deviation (in the sense of this work) is uniquely identifiable by a specific type of process pattern in the according analog wafer test datasets of the affected product.*

Note that, regardless of this claim of uniqueness, the pattern can occur in multiple electrical test parameters, i.e. on multiple wafermaps. This assumption is a prerequisite for this work, a discussion on its validity is provided at the end of this thesis in Section 4.6.5.

Pattern recognition is a common problem in machine learning, especially when handling image (or, as in our case, image-like) data. As explained in Section 2.2.2, wafer test data can be illustrated by wafermaps, which show a similar structure as single-channel images. By pattern recognition, specific pattern types shall be identified in such wafermaps. In contrast to the major part of image processing literature, the recognition comprises the whole wafermap, instead of

local regions or objects. We aim to assign each wafermap (as one instance) to one of the pattern types (classes). Hence, the problem can be interpreted as a classification problem in the sense of machine learning (or a clustering problem, if no labeled data is available and accordingly, the pattern types are not known a priori). Implicitly, these problem setups incorporate a second assumption:

Assumption 2 (Uniqueness of the pattern type)**.** *Each analog wafermap contains exactly one pattern, i.e. can be assigned to exactly one pattern type (including a pattern type of pure measurement noise, indicating the absence of any pattern on the wafermap).*

While Assumption 1 is required for the whole concept of the thesis, Assumption 2 can be loosened by applying a related technique - blind source separation - as an appropriate preprocessing methods, as will be explained in Section 2.3.2.

Based on these assumptions 1 and 2, the definition of a Health Factor is possible.

2.3.2 Blind-Source-Separation: A Related Problem

In literature, a related problem to process pattern recognition is blind source separation. Blind source separation basically tackles the problem of demixing linear combinations of signals. A typical example is provided by the cocktail party problem:

Different microphones m_i, $i = 1, \ldots, n_m$ record a discussion among n_s people, where s_j, $j = 1, \ldots, n_s$, denotes the signal obtained from the voices of person j (source j), $n_s \leq n_m$. Since no other source of noise is considered, each output signal m_i will be a linear combination of the inputs s_j with unknown weights $w_{i,j}$, i.e.

$$\begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_{n_m} \end{pmatrix} = \begin{pmatrix} w_{1,1} & w_{1,2} & \ldots & w_{1,n_s} \\ w_{2,1} & w_{2,2} & \ldots & w_{2,n_s} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n_m,1} & w_{n_m,2} & \ldots & w_{n_m,n_s} \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_{n_s} \end{pmatrix}. \tag{2.1}$$

In general, $\boldsymbol{m} = (m_i)_{i=1,\ldots,n_m}$ is known from observations, while both, $\boldsymbol{W} = (w_{i,j})_{i=1,\ldots,n_m \ j=1,\ldots,n_s}$ and $\boldsymbol{s} = (s_j)_{j=1,\ldots,n_s}$, are unknown. The goal is to estimate the source signals s_j from the mixture output m_i. In general, the problem is underdetermined unless additional assumptions are made.

For semiconductor industry, this problem can be adapted for different test parameters obtained from wafer test: assuming that the sources s_j correspond to the patterns contained in the dataset and the mixtures m_i are obtained as measurement values in the test, blind source separation can be applied. In literature, such approaches e.g. assume that patterns can be divided into those depicting unreliable devices as outliers (defect patterns), those depicting process variations (process patterns) and those depicting noise (noise pattern), see Zernig [33]. Compared to this approach, we will not distinguish between defect patterns (noise with outliers) and noise patterns in the present work, but rather focus on process patterns. However, a distinction will be made between process patterns and test patterns, which are not explicitly covered in most other works.

Major contributions to the topic of blind source separation for analog wafer test data were made on the so-called Independent Component Analysis (ICA) method, investigated in the context of wafer test data by Zernig et al. [39, 40] and Zernig [33]. ICA is a statistical method, based on the more prominent Principal Component Analysis (PCA) method, with the additional assumption

that the sources s_i are stochastically independent from each other. Exploiting this assumption by a transformation to an orthogonal space yields estimated sources $\hat{s}_j$, as well as a weight matrix $\hat{\boldsymbol{W}}$. ICA showed appropriate results, especially in case of highly correlated mixtures m_i, to demix process patterns from each other. Hence, ICA will be considered as a preprocessing tool in this work, in case that Assumption 2 is violated by multiple pattern types present on a single wafermap. The ICA method was further used in works by Turakhia et al. [41] and Batholomaeus et al. [42].

Another approach to blind source separation focusses on the decomposition of pass/fail or bin wafermaps into distinct patterns. For this purpose, Non-negative Matrix Factorization (NMF) is used. This approach is pursued in the context of wafer test data in different publications, see Schachtner [43], Schachtner et al. [44–46], as well as more recent work from Siegert et al. [47]. Basically, the idea in this approach is to decompose the wafer test dataset in an unsupervised way into a product of two non-negative matrices, which can be interpreted in a similar way as for the ICA method. However, these recent works could achieve uniqueness of the results, which is not trivially possible with the ICA, although approaches like multirun ICA reduce the impact of the randomized parameter initialization [48].

Although pattern recognition can benefit from suitable blind source separation methods, they do not sufficiently cover the problem setup for pattern recognition. For instance, recognizing variations of patterns (e.g. changes in size, etc.) is merely possible using appropriate feature extraction techniques. However, in order to fulfill Assumption 2, blind source separation can be a relevant preprocessing step for subsequent machine learning techniques.

2.3.3 Further Related Work

State-of-the-art approaches and related works exist for each of the different tasks, which are encountered in this thesis. In order to provide a clear, specific review of the literature related to the single tasks from a data-science perspective, related work sections are attributed to each section in Chapter 3. Here, an overview on the current state-of-the-art in semiconductor industry shall be provided.

Most data-driven approaches tackling the automated analysis of wafer test data concentrate on pattern recognition, without embedding this into a framework with other types of information. Early works in this respect reach back to the 1990s including the work by Kibarian and Strojwas [49], who present a statistical model to exploit such spatial correlations on the wafer surface. However, their work rather provides a descriptive spatial model than a full recognition setup. A full classification setup for wafer test data was described by Cresswell et al. [50], presenting a classifier for test patterns (note that in their case, *test patterns* referred to all kinds of patterns on the wafer, i.e. differs from the wording used in the present work). Their suggested method is based on a tree-structured object classification algorithm, deriving decision rules from wafer test data. Zinke et al. [51] introduced an early neural network approach to analyze wafer test data, embedded in a data analysis software for wafer bin maps and other related data sources. The target of this work is to enable a fast detection of root-causes after yield loss has occurred.

After 2000, triggered by the "revival" of research in machine learning and data science, the number of research works tackling classification problems exploded. At that time, image processing techniques were merged with machine learning approaches, applied on wafer test data e.g. by

Huang et al. [52] to detect analog anomaly regions (defect clusters) on wafermaps. For this purpose, they applied a nearest-neighbor clustering approach. Models from spatial statistics were merged with neural networks by Hsu and Chien [53], in order to present a suitable method to extract patterns from bin wafermaps. Compared to earlier works, their algorithm is also suitable for clustering, i.e. unsupervised learning. Another neural network approach was presented by Chen and Liu [54], who deploy so-called adaptive resonance theory networks. Support vector machines were applied for pattern recognition on wafermaps by Chao and Tong [55], as well as by Li and Huang [56].

In the recent years, works on pattern recognition in wafermaps include Wu et al. [57], who tackles the problem for large-scale and especially focussing on fast feature extraction. Yu and Lu [58] extend the related problem of defect detection on wafermaps using a spatial version of discriminant analysis. Both of these works enter the field of reproducible research, performing experiments on publicly available wafer test datasets, such as WM-811K [59]. A combination of sophisticated feature extraction and decision tree classifiers is presented by Piao et al. [60], again deploying the WM-811K dataset for comparison. Deep learning approaches were used by Tello et al. [61], who enrich their previously published regression network by a so-called deep-structured machine learning model. Finally, a very popular topology of deep neural networks for image data, the CNN, was applied to detect defect patterns by Nakazawa and Kulkarni [62], as well as by Kyeong and Kim [63] outlining a very good performance when tackling a multiclass classification problem.

Approaches specifically tackling the pattern recognition problem on wafermaps in an unsupervised way include those by Taha et al. [64] and Alawieh et al. [65]. While the first work proposes to perform segmentation into Voronoi regions by applying k-means clustering, the latter suggests a combination of singular value decomposition and hierarchical clustering for this purpose.

Although the presented research works indicate that large effort has been taken to enable and automatize pattern recognition in wafer test data, specific aspects were not investigated, yet. Firstly, all of these works are applied on pass/fail or bin wafermaps, i.e. cannot detect process patterns unless a yield loss occurs. Secondly, the supervised approaches (especially using neural networks) require large labeled datasets for training, which are hardly available in practice - especially in case of new or low-volume products, since the characteristics of pattern types vary from product to product. Furthermore, they cannot handle new pattern types. On the other hand, unsupervised methods are hardly compatible with expert-knowledge, since e.g. the characteristics of a specific cluster are not known a priori, which would be necessary to provide a process link via expert knowledge. As a third aspect, most of the approaches presented in literature are merely focussed on the pattern recognition task - although pattern recognition is also a major component of the Health Factor in the present work, it will be further combined with expert knowledge and pattern intensity, which is a novel aspect. This thesis provides new possibilities for the analysis of wafer test data by considering all of these aspects, i.e. analog wafermaps, new pattern types, few labeled training data, expert knowledge and pattern intensity.

2.4 Decision Support for Process Pattern Monitoring

Resuming to the assumptions made in Section 2.3.1, we aim to formalize the objective of this work, explained in Section 1.3, in a mathematical way. A key aspect for this purpose is the definition of a decision support system, which will be provided in short.

2.4.1 Decision Support Systems

A Decision Support System (DSS) is a framework to support humans in making decisions by data analysis, see e.g. Marakas [66] or Power [67]. Such systems are especially useful if large datasets are available, which contain relevant information on the situation and the possible actions. Commonly, a DSS is coupled with a knowledge based system, i.e. systems to extract essential information (knowledge) out of data. Driven by big data and the availability of incremental computational resources, DSS have grown in their capabilities and expanded their range from its roots in business and economics to other fields, such as IBM's Watson, entering the field of decision support for medical doctors [68].

Perceptions on the definition of a DSS vary between academia and business [69], but in this work, the scientific aspect will be pursued. Hence, a DSS is defined to be an automatized tool, which suggests a specific decision to the user, based on situation-specific information, as well as historical data and expert knowledge. In the sense of statistics, this situation is tailored for introducing Bayesian approaches, as will be explained in Section 3.4. In the field of data science, similarities can be observed between DSS and recommender systems, both sharing the goal to provide adequate suggestions to the user. However, while recommender systems aim to fit the needs of the user (e.g. suggest movies, music or shopping items), the DSS focusses on making an optimal decision on more objective circumstances.

A related topic is covered by decision making systems, deviating from a DSS (in the present definition) mainly by the fact that a decision is directly implemented by the system, while DSS present the decision to the user. The differences between decision-support and decision-making are rather application-relevant than technical. In fact, knowledge-based systems can be used for both, decision-making and decision-support, as well as the statistical framework of decision theory, which fits both needs.

In the field of automation in manufacturing, the wording of DSS is not widespread, yet. However, supporting decisions taken by human experts by automated data-processing systems is a clear need in order to leverage industry 4.0. In this work, the goal is to provide the definition of a data-based system, which fulfills the requirements for such a decision support system for process monitoring in manufacturing. In detail, the suggested Health Factor is an indicator of the presence of critical process patterns. This indicator can be expanded to a DSS by defining a threshold, s.t. an alert / an action is triggered if the Health Factor violates the threshold. While an alert is not yet considered as a full decision, the Health Factor delivers information to the expert, depicting the pattern, which was detected and which was responsible for the violation - according to Assumption 1, hence, the expert is provided with a decision support on where to detect the root-cause of the problem. Regardless from the expansion to "full" DSS, the main focus of this work is to define a valid Health Factor as an indicator based on the available data and expert knowledge.

In general, a DSS is hard to validate [70], as the impact of an action is difficult to quantify in most situations. While in business applications an evaluation of a DSS is commonly performed by cost-benefit analysis of the system or subjective analysis (including user-friendliness, etc.), our main goal in this work will not be to quantify the economic impact of the system. Hence, the evaluation of the system will be purely done on the correctness of the decision, without providing specific information on the company, product or market.

2.4.2 The Process Link

While an obvious idea would be to integrate root-cause analysis, i.e. the connection between the process deviation during frontend production and the process pattern occurring at wafer test, into the Health Factor pipeline, an issue occurs: the wafer test data does not intrinsically contain such information. In order to establish a process link for previously unseen patterns, new data sources must be integrated into the system. Hence, in case that merely wafer test data is used, the information that links the type of process pattern with the source of the process deviation must be provided extrinsically by an expert, lacking other alternatives at this point - an alternative would be the usage of inline production data, which will be part of future work. Further information, which requires expert judgment and impedes full automation is the discrimination between patterns: pattern types may vary strongly within their class, or form subclasses. In such a case, it is not possible to judge, based on wafer test data, whether or not both subclasses belong to the same pattern type. Equally, they could define two distinct pattern types. Such information must be provided by the expert.

Once trained, essential information can be retrieved from the suggested Health Factor: the indicator is able to recognize patterns presented at training time, which allows the expert to draw conclusions on the root-cause based on his knowledge from process history and from technical know-how. In this respect, it is crucial to combine two types of information:

- data-based knowledge extracted from the wafer test dataset, which contains the intrinsic information on process patterns and their intensity levels on single wafermaps,
- expert-knowledge introducing extrinsic know-how, which cannot be obtained from the dataset.

The development of a suitable combination of these two aspects is the key to the definition of the Health Factor.

2.4.3 Formalization of the Problem

In order to define the problem of obtaining a suitable DSS for process patterns in a mathematical sense, fundamentals of statistical decision theory will serve as a baseline. Statistical decision theory is a field, which is related to game theory, differing in the assumption that in decision theory, the best decision should be made based on evidence on a random *state of nature*, while in game theory, an optimized strategy against intelligent opponents (with or without collaboration) is pursued. Hence, decision theory is widely accepted as part of statistics, while game theory is strongly related to optimization.

In decision theory, see e.g. Parmigiani and Inoue [71], one assumes that a decision $d \in \mathcal{D}$ has to be taken, where $\mathcal{D}$ denotes the decision space (set of all possible actions). The impact of the decision depends on a (potentially multi-dimensional) random variable $\boldsymbol{\theta} \in \Theta$, where $(\Theta, \Sigma, \mathcal{P})$

defines a probability space in the sense of the axioms of Kolmogorov [72] with a sigma algebra Σ and a probability measure $\mathcal{P}$. In order to model the decision problem, a deterministic *loss* function $V : \Theta \times \mathcal{D} \to \mathbb{R}^+$ is assumed, describing the effect or cost, if a state of nature $\boldsymbol{\theta} \in \Theta$ occurs and a decision $d \in \mathcal{D}$ is taken.

From the perspective of optimization, V serves as a target function of an optimization problem, which has to be minimized among all $d \in \mathcal{D}$. However, as the state of nature $\boldsymbol{\theta}$ is non-deterministic, its uncertainty must be taken into account. In the sense of Bayesian statistics, the decision is taken according to the *risk* function $\mathcal{R} : d \in \mathcal{D} \to \mathbb{R}$, defined as the expected loss among all possible states of nature:

$$\mathcal{R}(d) = \mathbb{E}_{\boldsymbol{\theta}\in\Theta}\left[V(d,\boldsymbol{\theta})\right] = \int_{\boldsymbol{\theta}\in\Theta} V(d,\boldsymbol{\theta})\; d\mathcal{P}(\boldsymbol{\theta}). \tag{2.2}$$

In a Bayesian model, the state of nature is commonly described with a posterior probability distribution $\mathcal{P}_{\boldsymbol{\theta}|d}$, resulting in optimizing the Bayesian posterior expected loss

$$\mathcal{R}(d) = \mathbb{E}_{\boldsymbol{\theta}\in\Theta}\left[V(d,\boldsymbol{\theta})\right] = \int_{\boldsymbol{\theta}\in\Theta} V(d,\boldsymbol{\theta})\; d\mathcal{P}_{\boldsymbol{\theta}|d}(\boldsymbol{\theta}).$$

A decision is considered as optimal in case that the risk function $\mathcal{R}$ is minimized among all decisions $d \in \mathcal{D}$, i.e. the optimal decision d^* is obtained as

$$d^* = \arg\min_{d\in\mathcal{D}} \mathcal{R}(d). \tag{2.3}$$

In the formulation obtained from decision theory, the objectives for providing the claimed Health Factor can be summarized as follows:

- specifying an adequate loss function V, which states the effect if a certain state of nature $\boldsymbol{\theta}$ co-occurs with a decision d,
- identifying and modeling the random parameters defining the state of nature $\boldsymbol{\theta}$, i.e. the degree of uncertainty present, by a posterior probability measure $\mathcal{P}_{\boldsymbol{\theta}|d}$,
- solving the optimization problem provided in Equation 2.3 given the loss function V and the posterior probability measure $\mathcal{P}_{\boldsymbol{\theta}|d}$.

The stated target will be achieved by combining 3 major components, describing essential aspects for the Health Factor: pattern type, pattern intensity and pattern criticality. While the main focus will be put on the pattern type, investigated in Sections 3.1–3.3, intensity and criticality will be presented in Section 3.4, where also the Health Factor will be proposed as a combination of them, complying with the criteria stated above.

3 Data Science Methods

3.1 Data Preprocessing

The characteristics of the wafer test dataset, in specific the differences between wafermaps and real-world images were analyzed in Section 2.2. In order to process such a data format, data preparation and preprocessing methods are of high importance.

With regard to data structures, a wafer test dataset consists of a large table, containing one line of numerical values per device. The columns can be divided into those providing meta-information (x-, y-coordinates, wafer number, lot number, site of the ATE, test program and time, etc.), test parameters (one column for each distinct electrical test parameter) and test results (e.g. pass/fail, bins, etc.), see Tab. 2.1. In addition, PAT values can be contained, which are handled as test results as they are not measured, but calculated by statistical methods.

Note that the data format described above deviates from the typical data format of images: usually an image is stored as a (potentially multi-channel) matrix, where each entry corresponds to a pixel value, whereas wafer test data are stored as single-channel matrices, where each line corresponds to one device, i.e. one pixel on each wafermap depicts the according value of the device. Hence, each column provides the test values for the distinct positions present on a wafer. Since conversion from the one type to the other is possible (with minor restrictions), distinct methods will favor either the original wafer test data format (denoted as *list format*) or the image format (denoted as *matrix format*) for efficient calculation. While normalization and outlier detection steps can be performed in both formats, smoothing and feature extraction algorithms favor the matrix format, where spatial neighborhoods can be evaluated in a more efficient way.

In this chapter, the discussed topics are intended to provide a methodological overview on the respective areas, but specifically focus on the explanation of the developments achieved in the research for this thesis. Hence, the level of detail depends on the amount of original contribution rather than on the spectrum of the area.

3.1.1 Data Normalization

Data normalization is of high importance for all data science methods, since the weights are equalized among the variables by introducing a common scale. The term describes (usually linear) transformations, which are used to convert data from different sources to a joint scale. Especially in the present case of wafer test data, comprising test values at various data ranges, normalization is crucial to ensure comparability between the instances, which is necessary for pattern recognition. In general, multiple options exist for this purpose. Given that $\boldsymbol{x}\mathbb{R}^n$ denotes a

dataset, the literature survey by Shalabi et al. [73] compares the following three major approaches for data normalization:

- z-score normalization
$$\boldsymbol{x}_{norm} = \frac{\boldsymbol{x} - \text{mean}(\boldsymbol{x})}{\text{sd}(\boldsymbol{x})}, \tag{3.1}$$
- min-max normalization
$$\boldsymbol{x}_{norm} = \frac{\boldsymbol{x} - \min(\boldsymbol{x})}{\max(\boldsymbol{x}) - \min(\boldsymbol{x})}, \tag{3.2}$$
- normalization by decimal scale
$$\boldsymbol{x}_{norm} = \frac{\boldsymbol{x}}{10^k}, k = \arg\min_{l \in \mathbb{Z}} \left\{ |x_i| < 10^l \ \forall x_i \in \boldsymbol{x} \right\}. \tag{3.3}$$

When working with real-world datasets, it is useful to use robust estimators instead of the mean and the standard deviation of the dataset in order to reduce the impact of skewed distributions and outliers. For this purpose, the concept of PAT can be exploited for the purpose of normalization, leading to the robust z-score normalization, given by

$$\boldsymbol{x}_{norm} = \frac{\boldsymbol{x} - \text{median}(\boldsymbol{x})}{\text{iqr}(\boldsymbol{x})}, \tag{3.4}$$

where the Inter-Quartile Range (IQR) is defined as $\text{iqr}(\boldsymbol{x}) = q_{0.75}(\boldsymbol{x}) - q_{0.25}(\boldsymbol{x})$. $q_p(\boldsymbol{x})$ denotes the empirical p-quantile of the dataset $\boldsymbol{x}$. Other quantiles can be deployed for the denominator instead of the quartiles, see e.g. Zernig [33].

As an alternative to applying classical normalization methods to test data, it has proven useful to use the range defined by the specification limits, [LSL, USL], for data normalization (linear transformation by specification limits):

$$\boldsymbol{x}_{norm} = \frac{\boldsymbol{x} - \text{LSL}}{\text{USL} - \text{LSL}}. \tag{3.5}$$

However, note that the latter normalization method requires that specification limits are provided, which represent extrinsic information. Such information must be provided e.g. by expert knowledge.

All of these normalization methods share a joint goal: assigning a common range to wafermaps from different tests. While standardization (a method usually applied for Gaussian distributed data) and robust normalization center the distribution to 0 and scale by a variance parameter, all other methods aim to achieve a $[0, 1]$ scale for the data. However, due to the fact that single devices may violate the specification or PAT limits, the $[0, 1]$ range cannot be guaranteed for all devices.

Criteria for the selection of a suitable normalization method include the following:

- robustness w.r.t. (point) outliers, see Section 3.1.2,
- linearity, i.e.
$$\exists \alpha, \beta \in \mathbb{R} : \boldsymbol{x}_{norm} = \alpha \cdot \boldsymbol{x} + \beta, \tag{3.6}$$

method	robustness	linearity	hard boundaries	stability
z-score	no	yes	no	no
min-max	no	yes	yes	no
decimal scale	no	yes	yes	no
robust z-score	yes	yes	no	no
specification	yes	yes	no	yes

Tab. 3.1: Properties of distinct normalization methods for wafer test data.

- hard boundaries, i.e.

$$\exists m_1, m_2 \in \mathbb{R} : m_1 < (\boldsymbol{x}_{norm})_i < m_2 \ \forall i, \tag{3.7}$$

- stability of the parameters w.r.t. new data, i.e.

$$\boldsymbol{x} = \boldsymbol{x}^{old} \cup \boldsymbol{x}^{new} \Rightarrow (\boldsymbol{x}_{norm})_{1,\dots,i} = \boldsymbol{x}_{norm}^{old} \text{ for } i = \dim(\boldsymbol{x}^{old}). \tag{3.8}$$

Note that the computational effort is not considered a criterion for comparison, since all methods are uncritical in this respect. In short, the properties of the methods stated above can be summarized as depicted in Tab. 3.1.

Further normalization methods (especially non-linear ones) are available, but none of these fulfills all 4 criteria evaluated above. Thus, the more intuitive selection of linear methods will be favored. In detail, the robust z-score normalization method will be used in this work if not explicitly stated otherwise.

3.1.2 Outlier Detection

In addition to the choice of a common data scale, outliers and anomalies cause problems in further processing. While no uniform definition of outliers exists, a popular statement by Hawkins [74] states that outliers are observations deviating from other observations such that they are likely to originate from distinct mechanisms. In other words, *point outliers* can be interpreted from a probabilistic perspective as follows: given a density p_X of the data distribution, an outlier $x \in \mathbb{R}^l$ is defined as a data point, which is located in a low-density region, i.e.

$$\int_{\mathcal{B}(x,\varepsilon)} p_X(y)\ dy < p_0, \tag{3.9}$$

where $p_0 \in \mathbb{R}^+$ is a very low probability value, $\varepsilon > 0$ is a small radius and $\mathcal{B}(x,\varepsilon) = \{y \in \mathbb{R}^l : \text{dist}(x,y) < \varepsilon\}$ is a small region around x, given the distance metric dist(.,.).

Other types of outliers are contextual outliers (outliers w.r.t. the local rather than the global data distribution) and collective outliers (i.e. groups of outliers). In literature, a large variety of methods is presented to detect all types of outliers. Many of these deploy statistical tools to estimate parameters of the data distribution. As a result, the probability of the occurrence of a data point can be calculated.

One outlier detection method, which is especially well-established in semiconductor industry [36] is PAT: after calculating the PAT limits for the data distribution, each point violating these limits is categorized as an outlier. Although this approach implicitly assumes that the data distribution is Gaussian, which will not hold in many cases, PAT has also achieved valid results for other data distributions. In general, PAT is able to detect point outliers. However, an adapted version of PAT, where contextual outliers can be detected by evaluating PAT limits over local neighborhoods, exists.

Other outlier detection techniques, which do not require a parametric distribution, include density-based or angle-based methods. While the first estimates the probability density in a non-parametric way, the second evaluates the angle of the point under investigation to all other data points. In comparison, inner points will have a larger angle, while outliers (outer points) are assigned low angles. Other non-probabilistic outlier detection methods include decision trees, nearest-neighbor techniques, clustering, etc. In general, the related field of anomaly detection has attracted interest in the machine learning and data science community, where many sophisticated approaches are applied for this purpose.

In order to suit the needs for wafer test data, standard PAT methods will be applied by default in the following. In case that they are detected, outliers are treated as missing values and imputed by calculated values as described in Section 3.1.3.

3.1.3 Missing Value Imputation

Missing values (NA values) occur in wafer test data due to various reasons. The major root-causes for missing values in the datasets evaluated in this thesis are stated as follows:

1. **positions outside the boundaries of the wafer**
   This case results from the fact that images are rectangular, while wafers have a round shape. For specific methods, which cannot handle non-rectangular shapes, wafermaps need to be complemented to a rectangular by appending missing values at the boundaries.
2. **PCM structures**
   As outlined in Section 2.2.2, certain positions on the wafer are reserved for inline testing. No device is manufactured at such positions, resulting in a missing value on the wafermap. Such structures can be identified by their regular spatial distribution on the wafer.
3. **devices failing in previous measurements**
   In case that a device fails in an electrical test, subsequent tests are not performed in order to e.g. avoid damage of the prober. Such devices are scrapped afterwards. The according positions are shown as missing values in the wafermaps of the subsequent tests.
4. **devices classified as outliers**
   Although they might be within the specification limits, devices showing an anomaly behavior are detected as outliers and set to NA, as explained in Section 3.1.2.

Missing values distort the optical flow and hence, should be replaced by conforming values. In addition, several image processing techniques require full images and do not allow missing values, i.e. imputation is essential in those cases. As the aim of this thesis is to detect larger local or global patterns, rather than focussing on single-chip anomalies, imputing single missing values does not change the overall pattern, if the number of adjacent values imputed on the wafermap

do not exceed a reasonable extent. Otherwise, if many adjacent pixels are missing on a wafermap, the wafermap is excluded from further processing steps.

Basically, missing values can be categorized into 3 different groups [75]:

- **missing completely at random (MCAR)**
  The fact that a data point is missing does not depend on its value, nor on the values of the observed data. In our case, this is valid for missing values originating from positions outside the boundaries, as well as for PCM structures, where the actual performance of the (imaginary) device is independent from the fact that it is missing on the wafermap.
- **missing at random (MAR)**
  The fact that a data point is missing depends on the other observations, but not on the ground truth value of the missing point. For wafermaps, this will hold for missing values originating from devices, which failed in previous tests, in case that the electrical test, where the device failed, is statistically independent from the test depicted on the wafermap.
- **not missing at random (NMAR)**
  The fact that a data point is missing depends on its real (but unknown) value. This is most likely the case for missing values of devices, which failed in previous tests, if the test parameters are dependent, or for devices classified as outliers. In the latter case, the criterion is fulfilled by definition.

Distinct techniques for imputation of missing values are available. The most fundamental strategy is to replace missing values by a global, constant value, such as the global mean, median, minimum or maximum. While mean and median are preferred for MCAR data points, minimum or maximum will be more suitable for missing values originating from outliers. However, the decision on whether to use the one or the other is difficult.

More advanced strategies include the replacement of missing values by key values from their local neighborhood. For this purpose, a local neighborhood can be defined (by a regular grid of e.g. 4, 8, 12 or 24 neighbors or using kNN), followed by the calculation of the local key value. Such key values can, again, range from mean or median to minimum or maximum. Using local neighborhoods, the spatial structure of a pattern is more clearly preserved than with global replacements. Other techniques include the Multivariate Imputation by Chained Equation (MICE) method, see van Buuren and Groothuis-Oudshoorn [76], where imputation is performed in an iterative way to achieve more accurate results.

In addition, a technique from recommender systems is applicable to impute missing values: collaborative filtering. With this method, neighbors are specified in comparison with data points showing similar test values in the foregoing tests. The imputation value is obtained from the nearest neighbors.

By default, missing data are imputed with the median of their local neighborhood in this work.

3.1.4 Test Pattern Extraction

Further, the effect of test patterns discussed in Section 2.2.5 is another question to be solved during preprocessing. Since this topic is very specific to semiconductor industry, few related works exist. However, the topic was tackled in the work of Alagić [29], proposing the so-called Test Pattern Extraction (TePEx) algorithm.

Site information (i.e. the information, which site of the wafer prober was used to measure a specific device on the wafer) can be obtained from the wafer test dataset and must be exploited to identify test patterns. A site pattern occurs, if the according needle of the prober shows a systematic bias. Since minor offsets between the sites are common, centering is a straight-forward approach to eliminate simple types of site patterns: if S_i denotes all devices on a wafer, which are measured with site i, the centering step for a device $\boldsymbol{x}$ measured with site i is defined as

$$\boldsymbol{x}_{\text{site norm}} = \boldsymbol{x} - \text{median}(S_i). \tag{3.10}$$

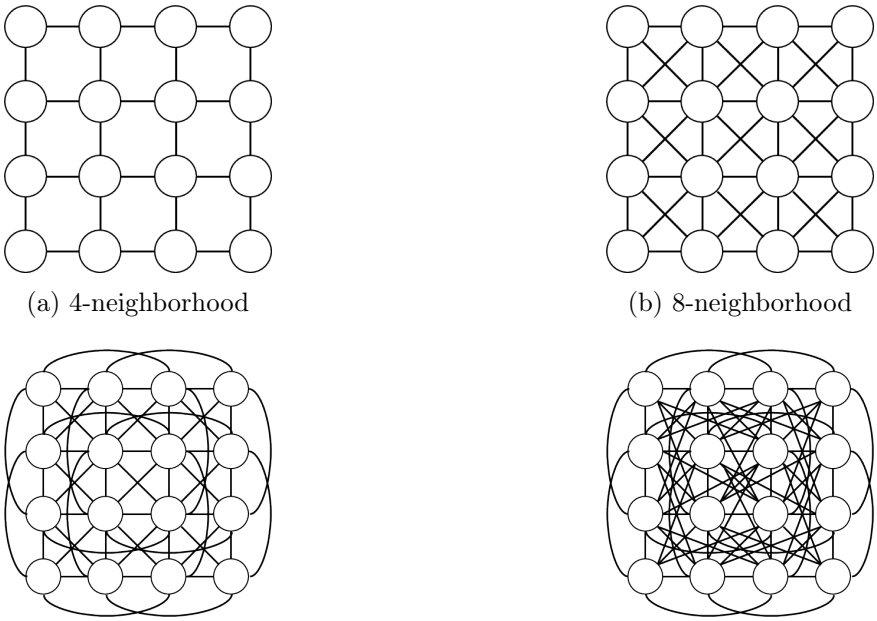
While constant offsets between the sites can be corrected via site centering, the work by Alagić [29] investigates a more challenging setup: it may happen that needles degrade over time during testing, e.g. due to contaminations. In this case, the correction via centering will not be able to remove the site effects. Instead, Alagić models each site as a stochastic process over the wafer, which can be interpreted as a time series.

After initial preprocessing steps including scale normalization by applying a first order difference operator, the autocovariance function (covariance between subsequent time points) is calculated from each time series to quantify the trends of the stochastic processes. Finally, it is necessary to judge whether or not the trends indicate that a prober needle degraded. For this purpose, a binning is applied to the autocovariance values in order to count the number of extraordinary high or low values. The TePEx value is then obtained from the Shannon entropy of these binned autocovariance values. Hence, it represents the degree of dependence within the devices measured with the same site: If a high entropy value is obtained, the time series is assumed to depict a strong trend, while low entropy values suggest an inconspicuous, random behavior.

With the values delivered by TePEx, sites showing deviations during testing can be identified. As a result, such site patterns can be handled in different ways: either the whole wafermap is neglected in further processing steps and scrapped, or a restoration of the values is pursued in order to preserve the global process pattern. In the latter case, all values from the affected site must be removed and replaced in the same way as missing values: these are filled with the median over the spatial neighborhood. However, this procedure is only reliable in case of highly parallel measuring, i.e. 8 or more sites. Otherwise, the proportion of scrapped devices is too high and imputation introduces a large bias.

3.1.5 Image Restoration: A Markov Random Field Approach

While data normalization, outlier removal, missing value handling and test pattern extraction are crucial preprocessing steps for analyzing wafer test data, an additional step is essential: wafermaps are commonly affected by measurement noise as well as spatial micro-structures, which can distort the subsequent pattern recognition algorithm. Hence, denoising methods are crucial for further processing steps. In specific, Markov Random Fields (MRFs) represent a sophisticated statistical framework to perform image restoration and denoising, especially tailored for semiconductor wafermaps in our previous work [13]. The explicit solution, which will be explained in the following, is part of the joint work with M. Pleschberger [14], as well as its extension in Pleschberger [77] - both works are associated with this project. The underlying model for image restoration originates from Geman and Geman [78] and is presented e.g. by Li [56].



(a) 4-neighborhood (b) 8-neighborhood

(c) 12-neighborhood (d) 24-neighborhood

Fig. 3.1: Different neighborhood structures for MRFs on regular grids.

Markov Random Fields From a statistical viewpoint, a random field is a multi-dimensional stochastic process, i.e. a family of random variables $\{X_{\boldsymbol{i}}\}$, where $\boldsymbol{i} \in S \subset \mathbb{N}^{\tau}$ is a multi-index. By definition, all random variables aggregated in the stochastic process are defined on a joint probability space $(\Omega, \Sigma, \mathcal{P})$. We assume that a wafermap is a realization of a 2-dimensional random field, hence the probability space comprises a continuous scale. An MRF is a stochastic process, which fulfills the so-called *Markov property* stating that the value of the MRF at each position $\boldsymbol{i}$ exclusively depends on the values of its first-order neighbors (i.e. vertices, which are directly connected by an edge in the graph), defined by a conditional independence assumption

$$\mathcal{P}(x_{\boldsymbol{i}}|x_{S\setminus\{\boldsymbol{i}\}}) = \mathcal{P}(x_{\boldsymbol{i}}|x_{\mathcal{N}(\boldsymbol{i})}) \ \forall \boldsymbol{i} \in S. \tag{3.11}$$

The set of first-order neighbors, $\mathcal{N}(\boldsymbol{i})$ is defined via a neighborhood graph G, which is commonly constructed by a circular radius around $\boldsymbol{i}$. On a regular grid (which is assumed for images), choosing a radius $r = 1, \sqrt{2}, 2$ or $2 \cdot \sqrt{2}$ results in a 4-, 8-, 12- or 24-neighborhood structure, respectively. These typical neighborhood structures are depicted in Fig. 3.1 for a 4×4 random field.

The Markov property does not only describe the local structure of the random field, but rather guarantees a global property of MRFs. Denoting the joint distribution of the random field by $\mathcal{P}_{\boldsymbol{X}}$, the Hammersley-Clifford theorem [79] states the following:

Theorem 1 (Hammersley-Clifford theorem)**.**
Given a random field $\{X_{\boldsymbol{i}}\}$*, it holds that*

$$\{X_{\boldsymbol{i}}\} \textit{ fulfills } (3.11) \Leftrightarrow \mathcal{P}_{\boldsymbol{X}} \sim f_{Gibbs}. \tag{3.12}$$

Hence, according to the Hammersley-Clifford theorem, the Markov property is fulfilled (local property) if and only if the joint distribution of the random field follows a Gibbs distribution (global property), defined by its density $f_{Gibbs} : \mathbb{R}^\tau \to \mathbb{R}$ as

$$f_{Gibbs}(\boldsymbol{x}) = Z(\beta)^{-1} e^{-\beta E(\boldsymbol{x})}, \tag{3.13}$$

where $Z : \mathbb{R}^\tau \to \mathbb{R}^+$ is a normalizing constant and $E : \mathbb{R}^\tau \to \mathbb{R}$ is a so-called energy function. In order to fulfill the requirements of the Gibbs distribution, the energy function E must be decomposable into clique potentials $V_l : \mathbb{R}^\tau \to \mathbb{R}^+$, i.e. functions assigning a positive value to each complete subgraph of G with size l. Hence, the energy function E can be decomposed as follows:

$$E(\boldsymbol{x}) = \sum_{\boldsymbol{i} \in C_1} V_1(x_{\boldsymbol{i}}) + \sum_{(\boldsymbol{i},\boldsymbol{j}) \in C_2} \sum V_2(x_{\boldsymbol{i}}, x_{\boldsymbol{j}}) + \sum_{(\boldsymbol{i},\boldsymbol{j},\boldsymbol{k}) \in C_3} \sum \sum V_3(x_{\boldsymbol{i}}, x_{\boldsymbol{j}}, x_{\boldsymbol{k}}) + \dots \tag{3.14}$$

As a result, these clique potentials represent the contribution of the single cliques C_i in the graph G to the joint distribution f_{Gibbs}. Given a 4-neighborhood structure, as depicted in Fig. 3.1a, the largest complete subgraphs in G have size $l = 2$, i.e. V_l is neglected for $l > 2$.

In the case of wafermaps, the random field is of dimension $\tau = 2$. In order to facilitate the notations, we will assume in the following that the pixels (which correspond to the devices on the wafer) are ordered in a column-wise manner, i.e. we reduce the multi-index $\boldsymbol{i} \in \mathbb{N}^\tau$ to a single index $i \in \{1, \dots, n\}$.

The image restoration model When applied for the purpose of image denoising, a popular model uses an MRF as follows: given a single-channel image $\boldsymbol{\omega} \in [0, 255]^n$, $n \in \mathbb{N}$, it is assumed that these data result from adding Gaussian white noise $\boldsymbol{\varepsilon} \in \mathbb{R}^n$ to the underlying true image $\boldsymbol{x}$. Here, $\boldsymbol{\varepsilon}$ and $\boldsymbol{x}$ are assumed to be stochastically independent. By definition, Gaussian white noise fulfills the property that

$$\boldsymbol{\varepsilon} \sim N(\boldsymbol{0}, \sigma^2 \mathcal{I}), \tag{3.15}$$

where $\sigma \in \mathbb{R}^+$ and $\mathcal{I}$ denotes the identity matrix.

In summary, we obtain the model as follows:

$$\boldsymbol{\omega} = \boldsymbol{x} + \boldsymbol{\varepsilon}. \tag{3.16}$$

The strategy to tackle this underdetermined system is based on a Bayesian procedure, hence it builds on the well-known Bayes' theorem, which is e.g. explained by Stuart and Ord [80]:

Theorem 2 (Bayes' theorem)**.** *Given random variables X and Y and according realizations x and y, where $\mathcal{P}(Y = y) \neq 0$, it holds that*

$$\mathcal{P}(X = x | Y = y) = \frac{\mathcal{P}(Y = y | X = x) \cdot \mathcal{P}(X = x)}{\mathcal{P}(Y = y)} \propto \mathcal{P}(Y = y | X = x) \cdot \mathcal{P}(X = x). \tag{3.17}$$

In the formula, the conditional probability distribution of $X|Y$ is called posterior distribution, while $\mathcal{P}(Y|X)$ is denoted as likelihood and $\mathcal{P}(X)$ is denoted as prior probability.

Concretely in the case of MRF, the likelihood results from the assumption stated for the distribution of the error, while the prior is obtained from the definition that $\boldsymbol{x}$ is a smooth surface. In detail, the likelihood is derived as

$$L(\boldsymbol{\omega}|\boldsymbol{x}) = \prod_{i \in S} \phi\left(\frac{x_i - \omega_i}{\sigma}\right), \tag{3.18}$$

where $\phi : \mathbb{R} \to \mathbb{R}$ denotes the density function of a standard normal distribution. From the structure of (3.18), it is obvious that the form of L is a special case of the Gibbs distribution in (3.13) with $V_l \equiv 0$ for all $l > 1$. In fact, the likelihood term L penalizes if the true image $\boldsymbol{x}$ deviates strongly from the observed image $\boldsymbol{\omega}$. As an energy function, the likelihood is expressed as follows:

$$\begin{aligned} E(\boldsymbol{\omega}|\boldsymbol{x}) &= \sum_{i \in S} \left(\frac{x_i - \omega_i}{\sigma}\right)^2 \\ &= \sigma^{-2}(\boldsymbol{x} - \boldsymbol{\omega})^T(\boldsymbol{x} - \boldsymbol{\omega}). \end{aligned} \tag{3.19}$$

In order to specify an adequate prior distribution, the smoothness of $\boldsymbol{x}$ must be taken into consideration. In mathematics, smoothness of a function $f : \mathbb{R}^n \to \mathbb{R}$ is commonly defined by the property that (for distance metrics dist_1 on $\mathbb{R}^n$ and dist_2 on $\mathbb{R}$)

$$\forall \epsilon \in \mathbb{R}^+ \exists \delta \in \mathbb{R}^+ : \ \text{dist}_1(\boldsymbol{x}, \boldsymbol{y}) < \delta \Rightarrow \text{dist}_2(f(\boldsymbol{x}), f(\boldsymbol{y})) < \epsilon \ \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n. \tag{3.20}$$

In accordance with this definition, smoothness is interpreted as the fact that two data points, which are sufficiently close to each other, have similar function values. However, the investigated MRFs are defined on a regular spatial grid, i.e. a discrete set as a function domain, which does not permit to define infinitesimally small environments. However, the basic idea of smoothness is adapted. Hence, observing an MRF, smoothness means that adjacent nodes (i.e. those, fulfilling the condition of $\text{dist}_1(\boldsymbol{x}, \boldsymbol{y}) < \delta$) have similar function values, i.e. $\text{dist}_2(f(\boldsymbol{x}), f(\boldsymbol{y})) < \epsilon$. In order to exploit this definition for defining a prior, which penalizes violations of the smoothness assumption, an intuitive choice is to set

$$\begin{aligned} E(\boldsymbol{x}) &= \sum_{\boldsymbol{i} \in S} \sum_{\boldsymbol{j} \in \mathcal{N}(\boldsymbol{i})} (x_{\boldsymbol{i}} - x_{\boldsymbol{j}})^2 \\ &= \boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}, \end{aligned} \tag{3.21}$$

where $\boldsymbol{M}$ is the adjacency matrix of the random field. Concerning its structure, this prior, again, fulfills the assumptions of a Gaussian distribution and hence, is a special case of the Gibbs distribution with $V_l \equiv 0$ for $l > 1$.

Due to Bayes' theorem (Theorem 2), the posterior distribution can be obtained by multiplying the likelihood and the prior distribution. Hence, in terms of the exponents, the posterior energy

$E(\boldsymbol{x}|\boldsymbol{\omega})$ results from adding the prior energy $E(\boldsymbol{x})$ and the likelihood energy $E(\boldsymbol{\omega}|\boldsymbol{x})$. Finally, the posterior energy has the following form:

$$\begin{aligned} E(\boldsymbol{x}|\boldsymbol{\omega}) &= E(\boldsymbol{\omega}|\boldsymbol{x}) + E(\boldsymbol{x}) \\ &= \sum_{i \in S} \left[\left(\frac{x_i - \omega_i}{\sigma} \right)^2 + \sum_{\boldsymbol{j} \in \mathcal{N}(\boldsymbol{i})} (x_{\boldsymbol{i}} - x_{\boldsymbol{j}})^2 \right] \\ &= \sigma^{-2} (\boldsymbol{x} - \boldsymbol{\omega})^T (\boldsymbol{x} - \boldsymbol{\omega}) + \boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x} \\ &\underset{\vartheta_0 = \sigma^{-2}}{=} \boldsymbol{x}^T (\vartheta_0 \mathcal{I} + \boldsymbol{M}) \boldsymbol{x} - 2\vartheta_0 \boldsymbol{x}^T \boldsymbol{\omega} + \vartheta_0 \boldsymbol{\omega}^T \boldsymbol{\omega} \end{aligned} \tag{3.22}$$

In order to gain the most information out of the posterior distribution, Bayes' estimators aim to select parameters, which maximize the posterior distribution. According to (3.13), where the Gibbs distribution is defined, it holds that

$$\arg\max_{\boldsymbol{x}} f_{Gibbs}(\boldsymbol{x}) = \arg\min_{\boldsymbol{x}} E(\boldsymbol{x}), \tag{3.23}$$

hence, the term in (3.22) is to be minimized. In the classical version of Li [81], gradient descent is suggested for this purpose, yielding severe runtime problems. As an alternative, the present work approaches the problem from an analytical perspective [14].

Due to the symmetry of the adjacency matrix $\boldsymbol{M}$ and the identity matrix $\mathcal{I}$, the gradient is derived as follows:

$$\nabla_{\boldsymbol{x}} E(\boldsymbol{x}|\boldsymbol{\omega}) = 2(\vartheta_0 \mathcal{I} + \boldsymbol{M})\boldsymbol{x} - 2\vartheta_0 \boldsymbol{\omega}. \tag{3.24}$$

For minimization, setting $\nabla_{\boldsymbol{x}} E(\boldsymbol{x}|\boldsymbol{\omega}) = \boldsymbol{0}$, we obtain

$$\boldsymbol{\omega} = \vartheta_0^{-1} \underbrace{(\vartheta_0 \mathcal{I} + \boldsymbol{M})}_{=:\boldsymbol{G}} \boldsymbol{x}, \tag{3.25}$$

which can be solved by Cholesky decomposition [82]. The according mathematical foundation is provided as follows:

Theorem 3 (Cholesky decomposition)**.** *Given a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times l}$, which fulfills the conditions that*

- *$\boldsymbol{A}$ is positive semi-definite, i.e. $\zeta_{\boldsymbol{A}} \geq 0$, $\forall \zeta_{\boldsymbol{A}} \ldots$ eigenvalues of $\boldsymbol{A}$, and*
- *$\boldsymbol{A}$ is Hermitian, a property which reduces to symmetry in the real-valued case.*

Then, there exists a lower triangular matrix $L \in \mathbb{R}^{m \times l}$ with positive diagonal entries, s.t.

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^T. \tag{3.26}$$

In case that $\boldsymbol{A}$ is positive definite, the decomposition is unique and the diagonal entries of $\boldsymbol{L}$ are strictly positive.

The condition of positive definiteness can be simplified to more fundamental criteria as follows (see e.g. Garcia and Horn [83]):

Theorem 4. *A matrix $\boldsymbol{A}$ is positive definite, if the following conditions hold:*

- *$\boldsymbol{A}$ is Hermitian (in case of real entries, symmetric),*

- *all main diagonal entries of* $\boldsymbol{A}$ *are positive and real,*
- $\boldsymbol{A}$ *is strictly diagonal dominant.*

In the case of the MRF model presented in (3.25), the matrix $\boldsymbol{G}$ fulfills these requirements for the following reasons:

- A matrix is positive semi-definite, if it is real, symmetric and strictly diagonal dominant.
- Both, $\boldsymbol{M}$ and $\mathcal{I}$ are symmetric, hence also $\boldsymbol{G}$ is symmetric.
- Strict diagonal-dominance is obtained from the following property, which is valid for all i:

$$\sum_{i \neq j} |\boldsymbol{G}_{ij}| = |\boldsymbol{G}_{ii}| \tag{3.27}$$

$$\begin{aligned} \sum_{i \neq j} |\boldsymbol{M}_{ij}| &= 2\vartheta \sum_{i \neq j} |\boldsymbol{G}_{ij}| \underset{(3.27)}{\leq} 2\vartheta |\boldsymbol{G}_{ii}| \\ &< 1 + 2\vartheta |\boldsymbol{G}_{ii}| = |1 + 2\vartheta \boldsymbol{G}_{ii}| = |\boldsymbol{M}_{ii}|. \end{aligned} \tag{3.28}$$

Alternatively, the properties of symmetry and positive definiteness can be deduced from the fact that $\boldsymbol{G}$ is the sum of the positive definite and symmetric matrices $\vartheta_0 \mathcal{I}$ and $\boldsymbol{M}$, since $\boldsymbol{M}$ is an adjacency matrix.

Apart from the fact that these conditions imply the existence of a unique solution of (3.25) according to the theory of linear algebra, they further permit to apply the Cholesky decomposition for solving the linear equation system in an efficient way. If the system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ has to be solved for an arbitrary vector $\boldsymbol{b} \in \mathbb{R}^n$, the Cholesky decomposition provided in Th. 3 divides the equation system into two sub-systems:

$$\begin{aligned} \boldsymbol{L}\boldsymbol{y} &= \boldsymbol{b} \text{ (forward substitution)}, \\ \boldsymbol{L}^T\boldsymbol{x} &= \boldsymbol{y} \text{ (backward substitution)}. \end{aligned} \tag{3.29}$$

As $\boldsymbol{L}$ is a triangular matrix, both systems can be solved in an efficient way by forward/backward substitution.

Due to the sparse block-diagonal structure of the matrix $\boldsymbol{M}$, an even more efficient way is provided by Block-Cholesky decomposition. Here, we exploit that $\boldsymbol{M}$ has the form

$$\boldsymbol{M} = \begin{pmatrix} \boldsymbol{A}_1 & \boldsymbol{B}_1^T & & & \\ \boldsymbol{B}_1 & \boldsymbol{A}_2 & \boldsymbol{B}_2^T & & \\ & \ddots & \ddots & \ddots & \\ & & \boldsymbol{B}_{\eta-2} & \boldsymbol{A}_{\eta-1} & \boldsymbol{B}_{\eta-1}^T \\ & & & \boldsymbol{B}_{\eta-1} & \boldsymbol{A}_\eta \end{pmatrix}, \tag{3.30}$$

where the number of blocks $\eta \in \mathbb{N}$ is determined by the number of rows of the wafermap due to the column-wise enumeration of the wafermap devices.

In analogy to the scalar case of the Cholesky decomposition, we set

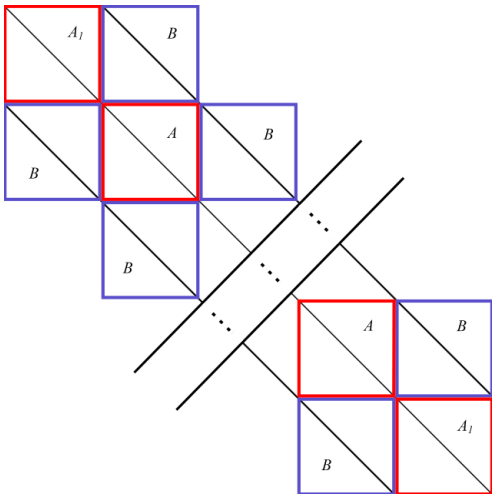


Fig. 3.2: Matrix structure of $\boldsymbol{M}$ to solve the MRF equation system explicitly. [14]

$$\boldsymbol{M} = \begin{pmatrix} \boldsymbol{L}_1 & & & & \\ \boldsymbol{C}_1 & \boldsymbol{L}_2 & & & \\ & \ddots & \ddots & & \\ & & \boldsymbol{C}_{\eta-2} & \boldsymbol{L}_{\eta-1} & \\ & & & \boldsymbol{C}_{\eta-1} & \boldsymbol{L}_\eta \end{pmatrix} \begin{pmatrix} \boldsymbol{L}_1^T & \boldsymbol{C}_1^T & & & \\ & \boldsymbol{L}_2^T & \boldsymbol{C}_2^T & & \\ & & \ddots & \ddots & \\ & & & \boldsymbol{L}_{\eta-1}^T & \boldsymbol{C}_{\eta-1}^T \\ & & & & \boldsymbol{L}_\eta^T \end{pmatrix} \tag{3.31}$$

and obtain the following equation system, which can be solved sequentially in an effective way:

$$\begin{bmatrix} \boldsymbol{L}_1\boldsymbol{L}_1^T &= \boldsymbol{A}_1 \\ \boldsymbol{C}_1 &= \boldsymbol{B}_1\boldsymbol{L}_1^{-T} \\ \boldsymbol{L}_2\boldsymbol{L}_2^T &= \boldsymbol{A}_2 - \boldsymbol{C}_1\boldsymbol{C}_1^T \\ \boldsymbol{C}_2 &= \boldsymbol{B}_2\boldsymbol{L}_2^{-T} \\ &\vdots \\ \boldsymbol{L}_{\eta-1}\boldsymbol{L}_{\eta-1}^T &= \boldsymbol{A}_{\eta-1} - \boldsymbol{C}_{\eta-2}\boldsymbol{C}_{\eta-2}^T \\ \boldsymbol{C}_{\eta-1} &= \boldsymbol{B}_{\eta-1}\boldsymbol{L}_{\eta-1}^{-T} \\ \boldsymbol{L}_\eta\boldsymbol{L}_\eta^T &= \boldsymbol{A}_\eta - \boldsymbol{C}_{\eta-1}\boldsymbol{C}_{\eta-1}^T. \end{bmatrix} \tag{3.32}$$

In case of a rectangular image, it holds that $\boldsymbol{A}_2 = \boldsymbol{A}_3 = \cdots = \boldsymbol{A}_{\eta-1}$, as well as $\boldsymbol{B}_2 = \boldsymbol{B}_3 = \cdots = \boldsymbol{B}_{\eta-2}$ due to reasons of symmetry. Furthermore, the matrix $\boldsymbol{M}$ is merely dependent on the neighborhood structure, but independent from the single measurement values. All $\boldsymbol{A}$-blocks (and all $\boldsymbol{B}$-blocks, accordingly) have the same matrix dimension. In detail, the matrix structure is depicted in Fig. 3.2

With respect to computational complexity, standard methods for matrix inversion require $O(n^3)$ operations and $O(n^2)$ units of storage, where n is the number of devices on the wafermap. Due to the sparse and block-diagonal structure, as well as the tridiagonality of M, the Block Cholesky decomposition reduces the number of operations to $\left(\frac{n}{k}\right) O(k^3) = O(n \cdot k^2)$ and the units of storage

# wafermaps	runtime [s] (GD)	runtime [s] (BCD)	rel. improvement
1	230	2.2	0.9904
100	23000	25	0.9989
1250	287500	210	0.9993

Tab. 3.2: Runtime comparison between gradient descent (GD) algorithm and the proposed explicit solution using a block Cholesky Decomposition (BCD) for the image restoration model using MRF. [77]

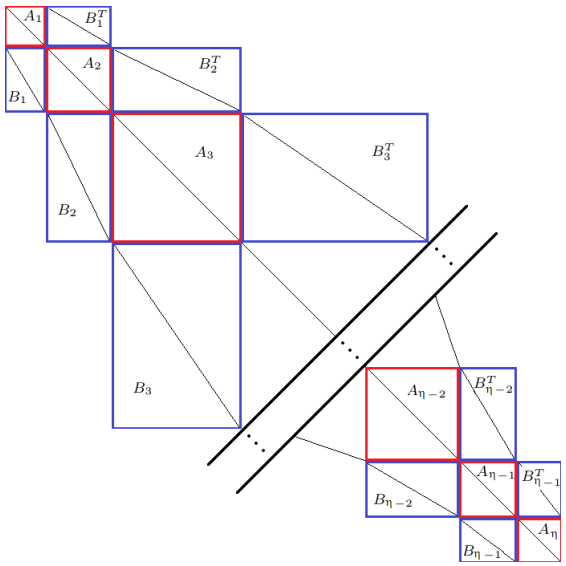


Fig. 3.3: Matrix structure of $\boldsymbol{M}$ to solve the MRF equation system for wafermaps.

to $\left(\frac{n}{k}\right) O(k^2) = O(n \cdot k)$, where k denotes the block size, $k \approx \frac{n}{\eta}$. Hence, the computational effort can be significantly reduced by deploying the beneficial structure of the matrices.

In particular, an empirical runtime comparison of the classical gradient descent algorithm and the presented analytical solution is presented in the experimental part of [14]. On ordinary images, a runtime improvement of approx. 90% could be achieved with the proposed variant. Although the evaluation might be subject to computational overhead and the implementations could be further improved, the benefit of the analytical solution is significant. However, the presented solution is only valid if a quadratic penalization function $E(\boldsymbol{x})$ is selected, otherwise (in case that e.g. discontinuities are considered) a numerical optimization procedure is required.

For wafer test data, where the size and structure of the wafermaps is equal within one product type, the improvement is even be higher. The runtime improvement for a sample wafer test dataset is depicted in Tab. 3.2 [77]. The comparison is performed for 1, 100 and 1250 wafermaps on the same hardware.

Adaptations for application on wafermaps Since wafers are not in a rectangular shape, the MRF model needs to be generalized: by definition, we iterate over the wafer in a column-wise order (from the perspective of runtime, a column-wise order is beneficial if the longest column is shorter than the longest row, otherwise a row-wise order should be preferred). With regard to

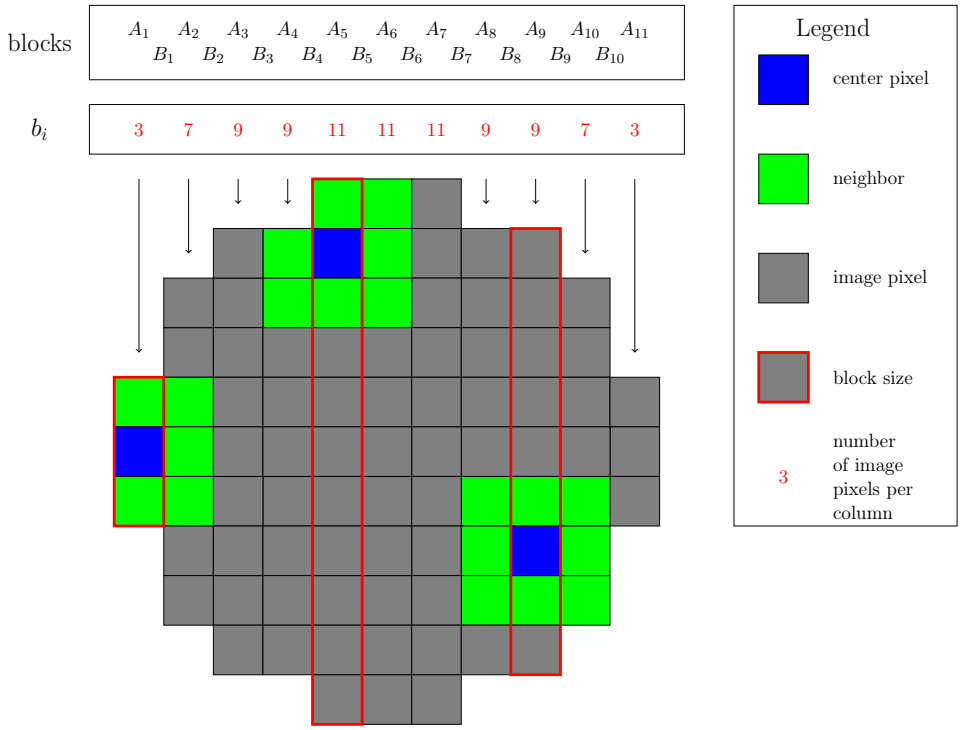


Fig. 3.4: Example of a wafermap, which produces distinct block sizes in $\boldsymbol{M}$ according to the column lengths for MRFs.

the neighborhood structure, the block size corresponds to the length of the rows, i.e. the matrix $\boldsymbol{M}$ consists of $\boldsymbol{A}$- and $\boldsymbol{B}$-blocks of distinct sizes. In accordance with Fig. 3.2, the structure of the matrix $\boldsymbol{M}$ in the MRF equation system for wafermaps is obtained as Fig. 3.3.

In general, the tridiagonal block structure of $\boldsymbol{M}$ is preserved, while the block sizes must be adapted. In detail, if b_i denotes the number of devices in the ith column of the wafer, the matrix $\boldsymbol{A}_i$ has the dimension $b_i \times b_i$, while $\boldsymbol{B}_i$ has the dimension $b_{i+1} \times b_i$. In general, the Block Cholesky decomposition remains valid and can be processed in an efficient way. However, more storage space is required for the blocks of $\boldsymbol{M}$, compared to the rectangular case, where regular structures can be exploited.

An example for the blocks obtained from a schematic wafermap is presented in Fig. 3.4, where the corresponding $\boldsymbol{A}$-blocks have the following sizes:

$$\begin{aligned}
\dim(\boldsymbol{A}_1) &= 3 \times 3 \\
\dim(\boldsymbol{A}_5) &= 11 \times 11 \\
\dim(\boldsymbol{A}_9) &= 9 \times 9 \\
\dim(\boldsymbol{B}_1) &= 7 \times 3 \\
\dim(\boldsymbol{B}_5) &= 11 \times 11 \\
\dim(\boldsymbol{B}_9) &= 9 \times 7.
\end{aligned}$$

3.2 Feature Extraction

Feature extraction is a central step of data mining, which consists of reducing the dimension of the processed data, such that relevant information is conserved, while unnecessary or redundant information is neglected. Hence, feature extraction is specific to both the data type and the problem - the latter defines the distinction between relevant and unnecessary information. Obviously, feature extraction is a crucial step in the data mining pipeline: if relevant information is removed, the subsequent machine learning algorithm will lack discriminative information. However, if unnecessary information is conserved, this might result in spurious correlations or interfere with the machine learning method.

Due to the image-like structure of the wafer test data, feature extraction methods from image processing can be considered. As only 1-dimensional values are observed for the wafermap, features from single-channel images will be extracted.

On the other hand, the problem setup defines specific requirements with regard to the chosen features. In particular, certain variations within a pattern type should be captured by the feature. Such variations include:

- position-invariance,
- scale-invariance (w.r.t. z-axis),
- scale-invariance (w.r.t. x-y-axis), i.e. resizing of the pattern,
- rotation-invariance.

Based on these criteria, the choice of suitable features is limited.

An alternative to engineered feature extraction is the usage of automated latent representations of the wafermaps, such as the latent space delivered by a Convolutional Variational Autoencoder [15]. However, due to the unsupervised nature of such approaches, variations can hardly be covered. For example, pattern A might require a rotation-invariant feature, while pattern B is specific to a certain position. The autoencoder will provide one common latent representation and, hence, cannot fulfil both properties simultaneously.

3.2.1 Image Feature Descriptors

In general, a large variety of image feature descriptors is available in scientific literature. Taxonomies of the field are provided in different surveys, e.g. in the work by Krig [84] or by Nguyen et al. [85]. Popular feature descriptors for image features are:

- texture-based features, e.g. Local Binary Pattern (LBP) [86],
- gradient-based features, e.g. Histogram of Oriented Gradients (HOG) [87],
- region-based features (blob detection), e.g. thresholding [88] or image segmentation [89],
- geometry-based features, e.g. Hough transform [90],
- color-based features, e.g. MPEG-7 Color Descriptors [91],
- shape-based features, e.g. Harris Corner Detector [92],
- keypoint-based features, e.g. Scale-Invariant Feature Transform (SIFT) [93].

In addition to these descriptors, a bunch of further, application-specific descriptors is available in literature, e.g. for face recognition. However, as only few investigations tackle the description of analog wafermaps via image features, there exists no specific category of methods for this domain, yet.

Due to the broad range of applications of image processing and computer vision, as well as the diversity of objects and shapes on images, the selection of an appropriate feature type is specific to the data characteristics and the problem. From the problem setup, an initial category selection can be performed by the following considerations:

- Process patterns on wafermaps are, by definition, regularities or regions of interest, rather than single-chip anomalies (outliers). As a consequence, outliers shall not influence the feature set - this contradicts the usage of keypoint-based methods, which are very sensitive to salient locations on the wafermap.
- Most pattern types result from inhomogeneously applying processes on the wafer. Hence, the wafer surface is subject to continuous variations over the x-y-axes rather than containing accentuated discontinuities. Therefore, on a discrete scale (which would be an according interpretation of the wafermap), the investigated patterns tend to approximate a smooth behavior as well. As a result, no succinct contours or edges are available on the wafermap, but rather gradual variations of the measurement values. Therefore, edge detection techniques will be excluded from further investigation.
- As deviations in real-world data hardly show the purity of geometrical structures, geometry-based approaches are not considered as well. Apart from the absence of contours or edges, which would be necessary to identify lines in the boundaries, even slight variations of a pattern would distort the geometric perfection and therefore, hinder recognition.

The main focus of the feature extraction step in this work will be put on gradient- or texture-based features, as well as the detection of an appropriate Region of Interest (ROI) (region-based features). From experimental results, especially two types of image features yielded accurate results on wafer test data: Histogram of Oriented Gradients (HOG) and LBP, together with its variant Rotated Local Binary Pattern (RLBP). The application of LBP and RLBP features is demonstrated in our associated publication [15].

3.2.2 Local Binary Patterns

The feature descriptor LBP, invented by Ojala et al. [86] based on previous work from He and Wang [94], is designed to extract texture information from an image. This texture information is mainly provided by the pixel values in the surrounding of each position. LBP features benefit from their robustness w.r.t. variations of the image, e.g. changes in the data scale. Several useful adaptations of the LBP method exist, including Uniform LBP [95] or RLBP [96].

Original LBP features In the fundamental version of LBP, a local neighborhood is defined for each pixel. Usually, for a position $\boldsymbol{i}$ on the wafer, a neighborhood $\mathcal{N}_{\boldsymbol{i}}$ is specified by a ring with center $\boldsymbol{i}$ and radius $r > 0$. All positions of the neighborhood, $\boldsymbol{j}$, are numbered clockwise by a function $\pi_{\boldsymbol{i}} : \mathcal{N}_{\boldsymbol{i}} \to \{0, \ldots, |\mathcal{N}_{\boldsymbol{i}}| - 1\}$. In order to obtain a description of the local neighborhood

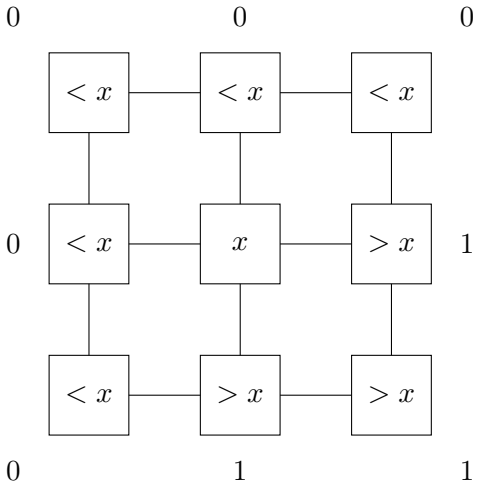


Fig. 3.5: Calculation of an $LBP_{\boldsymbol{i}}$ value for site $\boldsymbol{i}$ given an 8-neighborhood structure. In this case, $LBP_{\boldsymbol{i}} = (11100000)_2 = 224$.

of $\boldsymbol{i}$, LBPs specify a binary value $y_{\boldsymbol{j}} = \chi(x_{\boldsymbol{j}}; x_{\boldsymbol{i}})$ for each neighboring site $\boldsymbol{j}$ with the associated pixel value $x_{\boldsymbol{j}}$, i.e.

$$\chi(x_{\boldsymbol{j}}; x_{\boldsymbol{i}}) = \begin{cases} 1, & x_{\boldsymbol{j}} \geq x_{\boldsymbol{i}}, \\ 0, & \text{else.} \end{cases} \tag{3.33}$$

The LBP-value of position $x_{\boldsymbol{i}}$ is defined by

$$LBP_{\boldsymbol{i}} = \sum_{\boldsymbol{j} \in \mathcal{N}_{\boldsymbol{i}}} y_{\boldsymbol{j}} 2^{\pi_{\boldsymbol{i}}(\boldsymbol{j})}. \tag{3.34}$$

By definition, only elements with a full neighborhood are considered, which implies that $|\mathcal{N}_{\boldsymbol{i}}| = |\mathcal{N}_{\boldsymbol{k}}|$ for arbitrary positions $\boldsymbol{i}$ and $\boldsymbol{k}$ under investigation. Sites, which do not fulfil this property, e.g. those at the border of the image, are excluded from further evaluation. Thus, note that the LBP method requires a valid imputation of missing values as suggested in Section 3.1.3.

Considering the structure of the obtained $LBP_{\boldsymbol{i}}$ value, one obtains a decimal representation of the binary number $(y_{\pi_{\boldsymbol{i}}(0)}, \ldots, y_{\pi_{\boldsymbol{i}}(|\mathcal{N}_{\boldsymbol{i}}|-1)})_2$. Semantically, $LBP_{\boldsymbol{i}}$ represents the surroundings of site $\boldsymbol{i}$ on the image in a robust way, indicating e.g. gradient directions or edges.

Globally on the wafermap, the LBP feature descriptor is obtained by calculating a histogram of the $LBP_{\boldsymbol{i}}$ values for all sites $\boldsymbol{i}$, where each histogram bin corresponds to one integer value out of the domain $\{0, \ldots, |\mathcal{N}_{\boldsymbol{i}}| - 1\}$ for an arbitrary $\boldsymbol{i}$. In this work, the relative histogram will be used by default.

The special case of a ring-shaped neighborhood with $|\mathcal{N}_{\boldsymbol{i}}| = 8$ corresponds to an 8-neighborhood structure as introduced in Fig. 3.1 in the context of MRFs. For exemplary purpose, the calculation of $LBP_{\boldsymbol{i}}$ is demonstrated in Fig. 3.5.

In contrast to other methods, LBP features can be computed in an efficient way due to their simplicity. They also represent the surface structure of the image in a robust way w.r.t. the data scale. This property is crucial in the case of analyzing wafermaps, where the z-axis is basically unbounded (although normalization is performed, as described in Section 3.1). However, LBP

features might be heavily affected by measurement noise - a single outlier in the neighborhood might distort one digit in the binary LBP-representation and, as a consequence, result in a different $LBP_{\boldsymbol{i}}$ integer. In addition, LBP features are not invariant w.r.t. rotation of the image. The dimensionality of LBP is $\dim_{\mathrm{LBP}}(|\mathcal{N}_{\boldsymbol{i}}|) = 2^{|\mathcal{N}_{\boldsymbol{i}}|}$, which grows rapidly at an increasing neighborhood size $\mathcal{N}_{\boldsymbol{i}}$. The original version of LBP further suggests to divide the image into cells and concatenate the histograms calculated within each cell - hence, an even higher-dimensional feature vector is obtained. In this work, merely a global histogram of the LBP values will be applied. In order to overcome the limitations of lacking robustness w.r.t. noise, lacking invariance w.r.t. rotation and high dimensionality, a number of modifications of LBP are proposed, including Uniform LBP and RLBP.

Uniform LBP features Barkan et al. [95] developed a refined version of LBP, called Uniform LBP, which is tailored to enhance the robustness w.r.t. noise and to reduce the overall dimension $\dim_{\mathrm{LBP}}(|\mathcal{N}_{\boldsymbol{i}}|)$ of the LBP feature descriptor.

The idea of Uniform LBP is that those sites, where $LBP_{\boldsymbol{i}}$ shows many $0-1$- or $1-0$-transitions, are not informative for the description of the image content but rather indicate a noisy surface. The informative part of the image is hence assumed to be clearly oriented and therefore shows few transitions. The Uniform LBP features exclude each site $\boldsymbol{i}$, where $LBP_{\boldsymbol{i}}$ shows two or more $0-1$- or $1-0$- transitions in its binary representation, from the ordinary histogram bins and assigns them to an additional residual bin.

With regard to the dimension of LBP, the number of elements with less than two $0-1$- or $1-0$-transitions is derived from combinatorics: the number of binary vectors with no $0-1$-transition is 2, i.e. a vector consisting of zeros or ones, respectively. The number of vectors with one $0-1$-transition is $(|\mathcal{N}_{\boldsymbol{i}}|-1)$, obtained by permuting the position of the transition from the second to the $(|\mathcal{N}_{\boldsymbol{i}}|-1)$th position:

$$\left.\begin{array}{ccccccccc} 0 & 1 & 1 & 1 & \dots & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & \dots & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & \dots & 1 & 1 & 1 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 \end{array}\right\} |\mathcal{N}_{\boldsymbol{i}}|-1 \tag{3.35}$$

The same is valid for the number of vectors with exactly one $1-0$-transition. The number of such vectors with exactly two transitions, i.e. one $0-1$-transition and one $1-0$-transition is similarly obtained by the following construction:

$$\left.\begin{array}{ccccccc} 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & \\ 0 & 0 & 0 & 0 & \dots & 1 & 0 \end{array}\right\} \; |\mathcal{N}_i| - 2 \tag{3.36}$$

$$\left.\begin{array}{ccccccc} 0 & 1 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 1 & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & \\ 0 & 0 & 0 & 0 & \dots & 1 & 0 \end{array}\right\} \; |\mathcal{N}_i| - 3 \tag{3.37}$$

$$\vdots$$

$$\left.\begin{array}{ccccccc} 0 & 1 & 1 & 1 & \dots & 1 & 0 \end{array}\right\} \; |\mathcal{N}_i| - (|\mathcal{N}_i| - 1) = 1 \tag{3.38}$$

Hence, the number $c_{0-1,1-0}$ of vectors with one $0-1$-transition, followed by a $1-0$-transition equals

$$c_{0-1,1-0} = \sum_{j=2}^{|\mathcal{N}_i|-1} |\mathcal{N}_i| - j = \frac{1}{2}(|\mathcal{N}_i|^2 - 3|\mathcal{N}_i| + 2). \tag{3.39}$$

For reasons of symmetry, it holds for the number $c_{1-0,0-1}$ of vectors with one $1-0$-transition, followed by a $0-1$-transition that

$$c_{0-1,1-0} = c_{1-0,0-1}. \tag{3.40}$$

The total number such combinations, which is equal to the dimensionality of the Uniform LBP features $\dim_{\mathrm{ULBP}}(|\mathcal{N}_i|)$, is

$$\dim_{\mathrm{ULBP}}(|\mathcal{N}_i|) = 2 + 2(|\mathcal{N}_i| - 1) + (|\mathcal{N}_i|^2 - 3|\mathcal{N}_i| + 2) = |\mathcal{N}_i|^2 - |\mathcal{N}_i| + 2, \tag{3.41}$$

plus 1 for the residual bin (if considered).

In a typical setup, where the number of neighbors is set to $|\mathcal{N}_i| = 8$, one obtains that $\dim_{\mathrm{ULBP}}(8) = 59$ (with residual bin), which yields a strong benefit in contrast to the non-uniform case, where $\dim_{\mathrm{LBP}}(8) = 256$.

Despite this useful adaptation, it is not guaranteed that the information removed by Uniform LBP is negligible for all datasets and for all applications. Nevertheless, especially if unsupervised learning is concerned, dimensionality reduction is a necessary step in order to achieve appropriate results.

RLBP features While LBP features are sensitive to rotation of the compared images, Mehta and Egiazarian [96] suggested a rotation-invariant version of LBP, the so-called RLBP features.

While for LBP, the $LBP_{\boldsymbol{i}}$-value is obtained from its neighbors in a fixed, clockwise order, RLBP add an offset to the sequence during iterating over the neighbors of $\boldsymbol{i}$. In detail, the offset value $o_{\boldsymbol{i}}$ is defined as

$$o_{\boldsymbol{i}} = \pi_{\boldsymbol{i}} \left(\arg\max_{\boldsymbol{j} \in \mathcal{N}_{\boldsymbol{i}}} |x_{\boldsymbol{j}} - x_{\boldsymbol{i}}| \right). \tag{3.42}$$

The idea is to orientate the neighbors of $\boldsymbol{i}$ in a way that the iteration starts at the value with the largest difference to the value $x_{\boldsymbol{i}}$, i.e. the direction of the gradient (ascent or descent). The orientation is performed as follows:

$$RLBP_{\boldsymbol{i}} = \sum_{\boldsymbol{j} \in \mathcal{N}_{\boldsymbol{i}}} y_{\boldsymbol{j}} 2^{(\pi_{\boldsymbol{i}}(\boldsymbol{j}) - o_{\boldsymbol{i}})_{\text{mod } 8}}, \tag{3.43}$$

where $(.)_{\text{mod } 8}$ denotes the modulus operator (division by 8). The direction, which is assigned to the site $\boldsymbol{i}$ via $o_{\boldsymbol{i}}$ is called "dominant direction".

In general, the dimensionality of RLBP features equals to the according LBP setup, i.e. $\dim_{\text{ULBP}}(|\mathcal{N}_{\boldsymbol{i}}|) = 59$ in the uniform case and $\dim_{\text{LBP}}(|\mathcal{N}_{\boldsymbol{i}}|) = 256$ for $|\mathcal{N}_{\boldsymbol{i}}| = 8$ in the non-uniform case.

Adaptations for application on wafermaps The main advantage of LBP features (and their variants) for analog wafermaps is their robustness w.r.t. strictly monotonic transformations of the data scale. Hence, data normalization and rescaling becomes redundant. This is especially beneficial, since the data scale differs between wafermaps originating from different electrical parameters. More generally, LBP takes only ordinal information into account, which is beneficial in our case, since the absolute values of the measurement data are not of interest for pattern recognition. Furthermore, large datasets need to be processed, which requires that feature extraction can be performed in a computationally efficient way - another benefit of LBP.

However, additional aspects must be considered when analyzing and describing wafermaps by LBP features. These adaptations are explained and evaluated in our joint work with Santos et al. [15] and include:

- an additional image segmentation step to reduce background structure (i.e. pattern fragments, which overlay the investigated process pattern and result from other root-causes),
- the combination of LBP and RLBP, as rotation-invariant and non-rotation-invariant pattern types exist,
- the use of additional dimensionality reduction techniques, in order to obtain more compact features. The usage of Uniform LBP features is not sufficient in order to apply e.g. a clustering method on the feature space.

While the claim in Assumption 2 (Section 2.3.1) was that only one pattern type is present on each wafermap, relicts from minor deviations in other process steps that can be visible on the wafermap. In general, LBP and RLBP features are sensitive to such unimportant structures, which will be denoted as background of the wafermap. One way to reduce the background is to perform image segmentation. A trivial image segmentation method is thresholding, where each pixel value is compared to a threshold value $t \in \mathbb{R}$. In our case, all pixels above the threshold

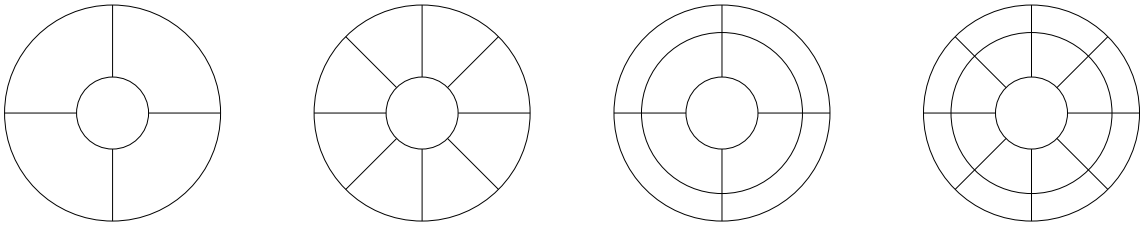
t are used, while pixels below the threshold t are set to 0. The main challenge is to select a suitable threshold t in an automated way. For this purpose, Otsu [88] presented an efficient heuristic: considering the pixel values above and those below the threshold as two classes, Otsu's method minimizes the intra-class variances, while - at the same time - maximizing the inter-class variance. Assuming a bimodal probability distribution of the pixel values, this procedure perfectly separates the two peaks. However, in a practical case with more or less than two mixture distribution components, the method can fail significantly. In this case, other image segmentation techniques, such as k-means clustering [97] will be a more appropriate choice, since they are able to segment a mixture of two Gaussian distributions in an optimal way, as well.

Resuming to the beginning of this section, there exist different types of process patterns with distinct characteristics and variations. A main challenge for feature extraction is the invariance w.r.t. rotation, which must be considered for some pattern types. For example, a pattern might be characterized by a spot on the border of the wafer, originating from a specific machine in the production process. However, it might happen that wafers are regularly plugged into the same machine in a rotated manner, hence the characteristic pattern will occur at another part of the wafer border. While in this case, a rotation-invariant method is desired, information on the rotation and the position is a characteristic of other pattern types and hence should be preserved. In order to resolve this problem, both types of information are conserved: the histogram obtained from LBP is concatenated with the histogram obtained from RLBP. In order to guarantee interpretability and comparability, the concatenated histogram is, again, normalized. The dimensionality of the feature vector doubles by this procedure.

While a feature vector of dimension 118, which is obtained from combining Uniform LBP and Uniform RLBP features, can be handled by some machine learning algorithms, most methods require further dimensionality reduction, especially when distance metrics are concerned (e.g. clustering). Although they differ in their invariance properties w.r.t. rotation, LBP and RLBP contain redundant information on the structure of the wafermap. Hence, linear correlations shall be eliminated in order to reduce the dimensionality - for this purpose, the PCA [98] algorithm is used, since it is able to rearranges the features linearly and orders the combinations w.r.t. their importance (judged by their contribution to the total variance of the dataset). Thereby, main information is extracted from the feature vector in an unsupervised way. In short, PCA performs a singular value decomposition of the covariance matrix of the feature components and transforms the feature matrix linearly to obtain axis-parallel eigenvectors, ordered by their corresponding eigenvalues. Dimensionality reduction is performed by removing feature combinations, which hardly contribute to the joint variance of the system, i.e. which correspond to low eigenvalues. With this procedure, the dimension of LBP and RLBP features can be drastically decreased.

3.2.3 Histogram of Oriented Gradients

HOG is a feature description technique, which is based on describing an image by a histogram, depicting the directions of the image gradient at each pixel. The idea was essentially coined by the work of Freeman and Roth [99], as well as the work of Dalal and Triggs [87]. In particular, the method achieved broad recognition for object detection, but (as in our case) can also be used for comparing regularities between images.



(a) $n_r = 1$, $n_\varphi = 4$ (b) $n_r = 1$, $n_\varphi = 8$ (c) $n_r = 2$, $n_\varphi = 4$ (d) $n_r = 2$, $n_\varphi = 8$

Fig. 3.6: Different image segmentations for the HOG method, including a distinct setting of the input parameters $n_r \in \{1, 2\}$ and $n_\varphi \in \{4, 8\}$.

HOG features When calculating HOG features, the image is divided into regular, rectangular blocks. At each position of these blocks, the image gradient is calculated, e.g. using the Sobel operator, first described by Duda and Hart [100]. From this operator, applied in x- and y-direction separately, the image gradient can be obtained at each pixel.

The next step is to bin the gradient directions in each block w.r.t. to predefined, equidistant bins. Based on this binning, a histogram is built for each block, weighting each data point by its gradient length. The histograms are normalized within each block before being concatenated among the blocks.

The dimensionality of HOG features depends on two input parameters: the number of blocks and the number of histogram bins. Clearly, a larger number of blocks will result in a more detailed description of the image, including also structures within small areas. However, apart from the increasing dimensionality of the obtained feature descriptor, its sensitivity to minor changes in the position of a pattern grows. On the other hand, the number of histogram bins effects the sensitivity of the method w.r.t. minor rotations of the pattern. For specific applications, both parameters are required to be in a reasonable range.

Adaptations for application on wafermaps In order to apply HOG to semiconductor wafer test data, several adaptations are necessary. The first aspect is the round structure of wafers (and also the circular shape of many pattern types), which has to be handled. Secondly, the input parameter selection must ensure that the dimensionality of the features do not exceed a reasonable range, but covers all main characteristics of the process patterns.

Concerning the circular shape of the wafer, a transformation of the coordinate system and a tailored version of defining the image blocks solves the problem: we transform the x-y-coordinate system (x, y) to polar coordinates (r, φ), where the point of origin is defined to be in the center of the wafer. As a result, neighborhoods are defined by the radius $r > 0$ and the angle $\varphi \in [0, 2\pi)$, which coincides with the structure of most pattern types on wafermaps. However, if a grid of rectangular blocks is defined on the transformated coordinate system, the cardinality of pixels in each block (block size) grows for increasing r due to geometrical reasons. Hence, especially the center of the wafer is split into small, nearly triangular segments, which can be resolved by defining a circular center region without any split w.r.t. the directions.

Instead of the number of blocks specified without polar-coordinate transformation, it is necessary to define the number of rings n_r (bins on the r-axis) and the number of segments n_φ (bins of

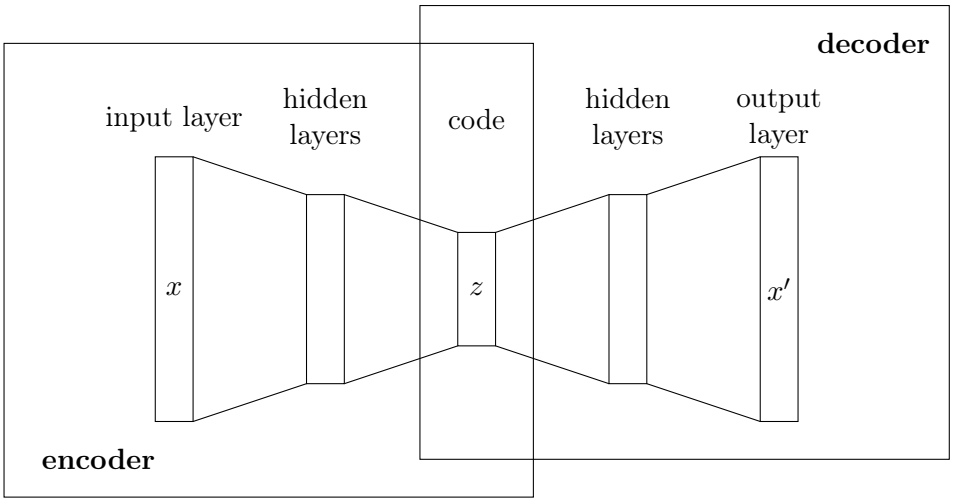


Fig. 3.7: The concept of an autoencoder network, where the encoded data representation (code) is extracted from the bottleneck of the network, cf. [101].

the φ-axis). An example of the wafermap-segmentation for HOG is provided in Fig. 3.6, where different parameter settings are depicted.

In total, the dimensionality $\dim_{\text{HOG}}$ of the HOG features obtained from the polar-coordinate representation depicted in Fig. 3.6, can be calculated as follows:

$$\dim_{\text{HOG}}(\lambda, n_r, n_\varphi) = \lambda \cdot (n_r \cdot n_\varphi + 1), \tag{3.44}$$

where λ denotes the number of histogram bins (binning of the gradient directions). Despite depending on the wafer and product size, the parameters will be selected in the range of $n_r \leq 3$, $n_\varphi \leq 12$ and $h \leq 10$ in the following, which implies that $\dim_{\text{HOG}} \leq 370$. Equivalently to the LBP approach, dimensionality reduction techniques can be applied for the HOG features.

3.2.4 An Alternative Setup: Convolutional Variational Autoencoders

As an alternative to the conventional image processing pipeline, where an application-specific selection of the image features is made by the expert, deep learning approaches offer feature selection and dimensionality reduction as well. One major class of representatives of such methods are autoencoders, an unsupervised neural network topology, which was introduced in the framework of deep learning by Liou et al. [102]. The basic principle of autoencoders is the following: in the first layers of the neural network, the so-called encoder part, a compression of information is provoked by reducing the number of available neurons from layer to layer until the intended dimensionality of the data is reached (bottleneck). Subsequently, the same network topology is appended in a reverse order (decoder part), i.e. a growing number of neurons from layer to layer until the original data dimension is reached. An example of an autoencoder topology is presented in Fig. 3.7. During training, large amounts of samples are fed into the network - the output of the network is then compared to the input, expressing the error term as the difference between these vectors with the goal to reconstruct the input at the back end of the network. The compressed, lower-dimensional representation of the data (code) at the bottleneck, corresponds to the features obtained from the network.

	supervised learning	unsupervised learning
discrete label set	classification	clustering
continuous label set	regression	dimensionality reduction

Tab. 3.3: Typical machine learning problems of the main categories, supervised and unsupervised learning with discrete or continuous label sets.

As explained e.g. by Kingma and Welling [103], as well as by Rezende et al. [104], convolutional autoencoders represent a combination of the two concepts, autoencoders and CNNs - in this case, the usage of convolution and pooling layers achieves a stepwise compression of high-level information from images, whereas the overall structure of the network follows the concept of autoencoders. Another extension of the autoencoder concept are variational autoencoders, i.e. autoencoders, where a distribution assumption is defined for the encoded variables. In this case, the parameters of this latent distribution are modeled. The combination of both concepts, convolutional autoencoders and variational autoencoders is called Convolutional Variational Autoencoder (CVAE) - this concept is a promising approach for feature extraction from wafermaps, which was demonstrated in the works of Santos et al. [15, 105].

In the experimental part of this work (Section 4), a comparison between texture-based LBP / RLBP features and CVAE will be made, assessing the benefits and drawbacks of a classical feature engineering approach introducing image processing techniques and a purely data-driven deep learning setup.

3.3 Machine Learning for Pattern Recognition

Machine learning comprises methods and algorithms, where decision rules are obtained from historical data (training) instead of being specified and implemented by an expert. Typical problems, which can be tackled by machine learning, are regression, classification or clustering. In all cases, a target variable (e.g. a class label) has to be modeled, assuming a statistical relation to certain features observed from the input data (dependent variables). The range of algorithms is broad, but can be categorized w.r.t. the type of meta-information available for the training data:

- **Supervised learning**: ground truth labels are known for all training samples, i.e. the values of the target variable can be used for training.
- **Unsupervised learning**: ground truth labels are unknown for all training samples.
- **Semi-Supervised learning**: ground truth labels are known for a part of the training data.
- **Reinforcement learning**: ground truth labels are unknown for the training data, but feedback can be provided for specific outcomes.

Apart from this taxonomy, categories of machine learning algorithms can be defined based on the data type of the target variable (such as algorithms specialized on time series or image data, e.g. CNN). Typical representatives of the main categories are presented in Tab. 3.3, although further problems exist, which are not covered in these groups.

Another discrimination of machine learning algorithms is made w.r.t. deep or shallow learning. Although not formally defined, deep learning refers to algorithms with a large number of model parameters, showing a high flexibility and the capability to involve very abstract, high-level information in the decision. Many examples for deep learning refer to topologies of artificial neural networks, such as CNNs, Deep Belief Networks, Long Short-Term Memory Networks or GaNs, see e.g. Goodfellow et al. [106]. Due to their high number of parameters, deep learning systems are specifically useful for big data problems, where large amounts of data are available, but cannot be trained efficiently by limited data. In this work, the focus will be put on shallow learning, since adequate, reliable training samples (especially in the supervised case) are limited in industrial manufacturing. Furthermore, a drawback of deep learning is, in general, that few insights into the learning procedure and latent data representation is possible, which leads to a low interpretability of the results.

In this section, appropriate machine learning approaches will be presented for pattern recognition in analog wafer test data, following the feature extraction steps presented in Section 3.2. Since pattern recognition might involve unsupervised, supervised as well as semi-supervised procedures, all three categories can be exploited for the final Health Factor.

3.3.1 Unsupervised Learning

Clustering is the most prominent representative of unsupervised learning, other problem setups include e.g. dimensionality reduction, anomaly detection, pattern mining or blind source separation. In terms of pattern recognition problems, the goal is to assign instances to distinct groups (or clusters) according to their mutual similarities. Hence, clustering is mainly based on an appropriate distance or similarity measure $\text{dist} : \mathbb{F} \times \mathbb{F} \to \mathbb{R}$, which fulfills in most cases the conditions of a metric from the viewpoint of mathematics. Based on the definition, a metric needs to fulfill 4 conditions, which are

1. $\text{dist}(\boldsymbol{x}, \boldsymbol{y}) \geq 0$, $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}$ (non-negativity)
2. $\text{dist}(\boldsymbol{x}, \boldsymbol{x}) = 0$ and $\text{dist}(\boldsymbol{x}, \boldsymbol{y}) = 0 \Rightarrow \boldsymbol{x} = \boldsymbol{y}$ $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}$ (definiteness)
3. $\text{dist}(\boldsymbol{x}, \boldsymbol{y}) = \text{dist}(\boldsymbol{y}, \boldsymbol{x})$, $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}$ (symmetry)
4. $\text{dist}(\boldsymbol{x}, \boldsymbol{z}) \leq \text{dist}(\boldsymbol{x}, \boldsymbol{y}) + \text{dist}(\boldsymbol{y}, \boldsymbol{z})$, $\forall \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathbb{F}$ (triangle equation)

However, there exist also similarity distance measures, which deviate from these conditions, e.g. pseudometrics, where a part of the definiteness condition, i.e. $\text{dist}(\boldsymbol{x}, \boldsymbol{y}) = 0 \Rightarrow \boldsymbol{x} = \boldsymbol{y}$ is not fulfilled. Other examples are the Kullback-Leibler divergence [107] (violating symmetry and the triangle equation), or the Cosine similarity (violating the triangle equation).

Distance measures Popular distance metrics in $\mathbb{R}^n$ are the $\mathcal{L}_p$-metrics, induced by the $\mathcal{L}_p$-norms: $\mathcal{L}_p(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|_p$, where

$$\|\boldsymbol{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}}. \tag{3.45}$$

The Euclidean metric ($p = 2$) is the most popular representative of this class. However, the Euclidean metric is known to perform bad in a higher-dimensional space, due to the curse of dimensionality. In this case, the $\mathcal{L}_1$-norm is a valid alternative, but the lack of continuous differentiability is not compatible with many algorithms.

Apart from $\mathcal{L}_p$-metrics, the Kullback-Leibler divergence can be transformed to a symmetric distance measure between probability distributions. In $\mathbb{R}^n$, the Kullback-Leibler divergence [107] is applicable, if $\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i = 1$ and $x_i, y_i \in [0,1]$, $\forall i \in \{1,\dots,n\}$, and is defined as follows:

$$D_{KL}(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{n} x_i \log\left(\frac{x_i}{y_i}\right), \tag{3.46}$$

given that $y_i > 0$. A symmetric measure dist_{KL} can be obtained from this term as follows

$$\text{dist}_{KL}(\boldsymbol{x},\boldsymbol{y}) = D_{KL}(\boldsymbol{x},\boldsymbol{y}) + D_{KL}(\boldsymbol{y},\boldsymbol{x}) \tag{3.47}$$

A related concept to Kullback-Leiber is the Jensen-Shannon divergence D_{JS}, which is defined as follows:

$$D_{JS} = \frac{1}{2}\left(D_{KL}\left(\boldsymbol{x}, \frac{1}{2}(\boldsymbol{x}+\boldsymbol{y})\right) + D_{KL}\left(\boldsymbol{y}, \frac{1}{2}(\boldsymbol{x}+\boldsymbol{y})\right)\right). \tag{3.48}$$

According to Endres and Schindelin [108], the Jensen-Shannon divergence can be transferred to a metric dist_{JS} by $\text{dist}_{JS}(\boldsymbol{x},\boldsymbol{y}) = \sqrt{D_{JS}(\boldsymbol{x},\boldsymbol{y})}$.

An adapted version of the Jensen-Shannon metric for wafermap comparison Basically, all of the presented distance metrics can be deployed to compare images and therefore also wafermaps, given that the features are on a normalized scale (i.e. fulfilling the requirements of a probability distributions). However, due to high dimensionality, $\mathcal{L}_p$ norms have large difficulties on these datasets.

Furthermore, as explained in Section 3.2.2, it proves useful to consider extracting information from ROIs of the wafermap. In this case, regions need to be compared to each other instead of equally-sized images, i.e. the size of the region must be taken into account for the comparison. For this setup, Geiger et al. [20] presented a valid concept based on previous work by Geiger [109] to adapt the Jensen-Shannon divergence for features extracted from wafermap regions, considering the region size: if two wafermaps $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are characterized by their features $\boldsymbol{f}_1 \in \mathbb{F}$ and $\boldsymbol{f}_2 \in \mathbb{F}$, calculated over regions with the (relative) sizes $s_1 \in \mathbb{R}$ and $s_2 \in \mathbb{R}$, the distance can be quantified by the version of the Jensen-Shannon distance for wafermaps:

$$\text{dist}_{JS}(\boldsymbol{w}_1,\boldsymbol{w}_2) = D_{JS}\left((s_1, 1-s_1),(s_2,1-s_2)\right) + \frac{s_1+s_2}{2} D_{JS}\left(\frac{s_1}{s_1+s_2}\boldsymbol{f}_1, \frac{s_2}{s_1+s_2}\boldsymbol{f}_2\right). \tag{3.49}$$

The intention behind this formula is to weight the probability distributions covered by $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ by the region sizes s_1 and s_2 of their according regions. While the first term penalizes large differences between the compared regions, the second term penalizes differences between the contents of the regions (represented by their features $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$).

Clustering algorithms Among the large number of popular clustering algorithms, we will concentrate on two representatives of distance-based methods, hierarchical clustering and k-means clustering. Both of these algorithms rely on a distance metric, where k-means is specialized on the Euclidean metric since it operates on Euclidean spaces (although, it implicitly does not calculate pairwise distances but rather distances between a point and a centroid).

Hierarchical clustering calculates a pairwise distance matrix upon all data points. Then, either most similar points are sequentially aggregated (agglomerative clustering) or most distant points are sequentially separated from each other (divisive clustering), which can be represented by a tree, represented by a dendrogram. During the process, it is necessary to calculate distances from single instances to a cluster - for this generalization of distance measures, popular concepts define the distance between clusters as the shortest (single linkage), average (average linkage) or longest (complete linkage) pairwise distance between instances from cluster A and instances from cluster B. Given a predefined number of clusters, the tree is cut at an adequate threshold. Hierarchical clustering is deterministic, however, it requires large processing effort to calculate all pairwise distances, especially in case of large datasets. However, once the pairwise distance matrix is calculated, the result can be easily transferred to a distinct number of clusters and also scalability is possible, if new datapoints are added to an existing dataset.

Using k-means clustering, k center points (centroids) representing the clusters are introduced, such that the distances from each data point to their closest centroid indicates the assignement of the data point to the according cluster. While in the initial step, all centroids are randomly distributed in the feature space, sequential use of this cluster assignment, followed by a rearrangement of the centroids leads to convergence (although it is not guaranteed to find the optimal solution) [110]. Due to the randomized selection of the initial centroids, k-means clustering is non-deterministic. In contrast to hierarchical clustering, k-means performs better from a computational perspective (linear complexity can be achieved under specific circumstances), but depends on a fortunate selection of the initial elements (different heuristics exist for this purpose). Furthermore, k-means clustering is a representative of the group of prototype-based clustering algorithms, where each cluster is represented by one instance from the feature space (i.e. the centroid).

Other groups of clustering algorithms include angle-, distribution- or density-based clustering algorithms (e.g. DBSCAN), which utilize geometric properties, distribution assumptions or density regions in the data distribution to perform clustering.

Properties of clustering in pattern recognition Clustering shows two major drawbacks compared to supervised approaches for the purpose of performing pattern recognition: firstly, it is hard to introduce extrinsic information into clustering, i.e. an expert is not able to specify the type of result he wants to obtain. Using supervised or semi-supervised methods, this information is provided by the labeled training data.

Secondly, clustering algorithms are harder to evaluate. In contrast to classification results, which can be compared to their ground truth class assignment using an F measure, measures to evaluate clustering are e.g. glsnmi or Average Silhouette Coefficient (ASC). However, comparisons using Normalized Mutual Information (NMI) require ground truth information on the grouping of the data samples, which is not available in a reliable format in most cases. Furthermore, ASC

intrinsically quantifies the quality of clustering by the certainty of the assignments of the data samples, which is no extrinsic measure. Especially for recognition, it is necessary to judge whether the investigated object is correctly detected by the algorithm or not.

When analyzing wafer test data, where a high-dimensional feature space is observed, clustering results are heavily affected by spurious correlations, which hardly contain relevant information. Furthermore, since patterns show distinct invariance properties, all possible characteristics must be covered in the image features. Hence, a purely unsupervised approach such as clustering will not be able to achieve a sufficient quality for analyzing wafer patterns.

3.3.2 Supervised Learning

Supervised learning assumes that a labeled training dataset is available, i.e. the correct results are known for each training instance. When performing pattern recognition, classification is of major interest, since specific classes of patterns shall be recognized.

Classification algorithms comprise different approaches, some of which were evaluated against each other in previous work [16]. In the area of shallow learning, decision trees and random forests, as well as Bayesian procedures and SVMs are popular representatives - these approaches are summarized e.g. by Bishop [111] and explained in this chapter. In the field of deep learning, again, artificial neural networks are applicable. Furthermore, a distinction is made between generative and discriminative classifiers - while the first models the likelihood and the prior knowledge separately (i.e. builds a model for the underlying generating process of the data), the latter models the posterior distribution directly. In general, generative setups are preferable since they reveal information on the generating process of the data - however, more accurate results are often achieved by discriminative approaches.

In this work, specific focus is put on methods, which comprise few parameters and hence, can be trained using a limited amount of labeled data. Hence, deep learning is not taken into consideration, since a sufficient number of reliable labeled training samples cannot be provided.

Decision trees and random forests Firstly, decision trees are investigated: based on elementary decision rules (thresholding w.r.t. single features), a tree structure is constructed from the training data. Decision trees are discriminative classifiers and can be interpreted as a sequence of if-else conditions in terms of classical programming tasks, which are extracted in an automated way. Decision trees benefit from their transparency and their interpretable structure, i.e. a decision can be traced back to single informative features. However, no probabilistic model is used, hence no information on the underlying data distribution is gained. Decision-trees can be used for both, binary and multiclass problems.

An extension of decision trees is the random forest, i.e. an ensemble of mutually independent decision trees, each trained on a randomly sampled subset of the training data. Each single tree provides one vote for the selected class, the final classification is based on a majority vote. Random forests avoid overfitting by sampling from the training data and provide more insight into the data distribution. However, they require more training effort than decision trees. In general, the obtained trees of the random forest have a simpler structure than those obtained from a single decision tree on the same training set, since smaller subsets are used when applying random forests.

Logistic regression An approach, which is based on classical statistical models is logistic regression: while simple regression models provide a continuous output (target variable), a logistic regression model is a generalized linear model with a logistic function as link function, i.e. mapping the continuous output of a regression model to the interval $[0, 1]$. Hence, the result of logistic regression can be interpreted as a probability score for one of the classes. Logistic regression is restricted to a binary (2-class) problem, but an extension for the multiclass case exists [112].

A limitation of logistic regression arises from the high number of parameters: each feature is, by default, introduced into the model (unless feature selection is used a priori) - hence, the number of model parameters is at least as large as the number of features. In case that a high-dimensional feature space is observed, it is beneficial to restrict the model to the *most informative* features. In terms of linear models, the most informative features are those, which are attributed to model parameters deviating significantly from zero. Using regularization techniques as a prior assumption, the norm of the model parameters is penalized, leading e.g. to a sparse model in case of the $\mathcal{L}_1$-norm. Hence, the number of parameters deviating from 0 (and, hence, the number of features contributing to the model) is reduced. This procedure is called LASSO [113]. In case that regularization is performed using the $\mathcal{L}_2$-norm, ridge regression is obtained. The combination, where both $\mathcal{L}_1$- and $\mathcal{L}_2$-regularization is performed, exists under the term elastic net [114].

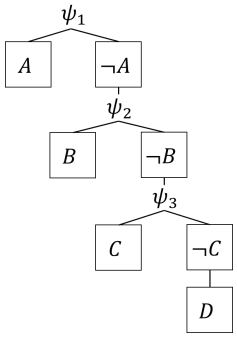
Regularization methods are closely related to the SVM classifier: Zhou et al. [115] showed that elastic net is a special case of a linear SVM. Similar results are obtained for LASSO, which is a special case of the elasic net.

Bayes classifiers From a generative perspective, classifiers based on Bayesian statistics allow a detailed insight into the process underlying the data. On the basis of this information, probabilistic statements and conclusions can be drawn. A Bayes classifier interprets the training data as realizations of a likelihood and combines it with prior knowledge, e.g. meta-information on the training samples. Since a parametric statistical model requires a distribution assumption, as well as covariance structures between the single features (variables), a high number of parameters needs to be covered.

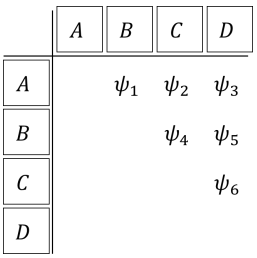
A simple representative of Bayes classifiers is the so-called Naive Bayes classifier. In this model, independence between the variables is assumed, hence, the likelihood model reduces to an ensemble of one-dimensional components. A typical assumption for continuously distributed data is the normal distribution, which will be used in the following: a variable x_i is modeled by a one-dimensional normally-distributed kernel $x_i \sim N(\mu_i, \sigma_i^2)$, hence the likelihood model for a single training data point x is obtained by

$$x \sim \prod_{i=1}^{d} N(\mu_i, \sigma_i^2). \tag{3.50}$$

Although providing surprisingly good results, more general Bayes classifiers, comprising dependencies between the features or more complex distribution assumptions, are in use. Bayes classifiers can be used to solve binary, as well as multiclass problems.



(a) One-vs-all classifier



(b) One-vs-one classifier

Fig. 3.8: Concepts of the one-vs-all and one-vs-one classification technique, where the multiclass problem with $n = 4$ classes A,B,C and D is resolved by decomposition into binary classifiers ψ_i.

Support vector machines The linear SVM is a discriminative classifier, based on the geometric idea of separating the classes in the feature space by a one-dimensional object (line, plane, hyperplane), such that the distance between the object and the data points is maximized. In contrast to classical maximum margin classifiers, SVMs allow data points to be on the wrong side of the object - hence, SVMs are powerful classification techniques, which can be trained by a limited amount of training data. Based on the distance measure in the feature space, the classifier can deliver a probabilistic output (soft classification) and cannot be used for multiclass classification. Various extensions to multiclass-setups or for more general problem types (semi-supervised problems, one-class problems, etc.) exist.

In order to generalize the concept of SVM to non-linear classification, kernel functions are in use. Applying the so-called kernel trick, the original data points are transformed in a non-linear way before introducing the separating hyperplane between the classes. Kernels can be of various types. Common choices are polynomial, Gaussian or hyperbolic tangent types of kernels.

Decomposition techniques While e.g. decision trees are intrinsically able to cope with multi-class problems, logistic regression or SVM must be adapted. However, concepts to reduce the multiclass problem to binary ones exist. Such decomposition techniques include one-vs-all, one-vs-one (round robin) and Error-correcting output coding (ECOC) classification, see [116,117]. In general, any type of binary classifier can be used as elementary classifier for these decomposition techniques.

One-vs-all classification follows the intuitive approach to train a binary classifier for each class, evaluating whether the object is in the respective class or not. Such decisions are taken by the binary elementary classifiers in a tree-based manner. An exemplary case with classes $A, \ldots, D$ is depicted in Fig. 3.8a. In the evaluation phase, the leaf of the tree, which the instance is assigned to, determines the class. A limitation of one-vs-all classification is given by the inhomogeneity of the "negative" class, i.e. in step one, the set of all classes except for class A - hence, many binary classifiers have difficulties when being trained with samples from different classes, labeled as instances from this negative class. In total, $n - 1$ binary classifiers need to be trained.

Another option is to apply one-vs-one classification, where all pairs of classes are separated by binary classifiers. Each of these provides a vote for the class the sample is assigned to - in total,

	ψ_1	ψ_2	ψ_3	ψ_4	ψ_5	ψ_6	ψ_7
c_1	1	1	1	1	1	1	1
c_2	0	0	0	0	1	1	1
c_3	0	0	1	1	0	0	1
c_4	0	1	0	1	0	1	0

Tab. 3.4: Error-correcting output codes to perform multiclass classification for $n = 4$ classes. [16]

the instance is assigned to the class with the highest number of votes. Here, the inhomogeneity of the negative class, which represented a drawback of one-vs-all classification, is resolved. However, the number of pairwise comparisons and hence the number of binary classifiers is $\frac{n(n-1)}{2}$, resulting in intractable computational effort for a larger number of classes. The concept of one-vs-one classification is demonstrated in Fig. 3.8b.

An alternative method to obtain a decomposition of the multiclass problem is ECOC. For this method, a binary code of fixed length is generated based on the number of classes, see e.g. Tab. 3.4, which indicates the combinations of classes to be distinguished by binary classifiers. For a new instance, an individual code is generated by evaluating all classifiers, which can be compared to the codes of each class - the instance is assigned to the class with the lowest Hamming distance. Since the code is not minimal (i.e. redundant), minor errors in the classification can be corrected by the method. However, the more redundancy is introduced, the more the number of binary comparisons grows. Exhaustive codes can be generated in an algorithmic way and comprise a code length of $2^{n-1} - 1$, which is equal to the number of binary classifiers trained. The degree of redundancy is indicated by the difference to the minimal code covering all classes, which has length $\lceil \log_2(n) \rceil$.

Properties of classification in pattern recognition For pattern recognition, classification shows complementary properties compared to clustering: by providing labels for the training data, a separation of the classes is possible on the basis of extrinsic information - hence, an expert can, for instance, interfere during training by actively providing specific training samples.

Compared to the evaluation of clustering, which is mainly based on the NMI value in our case, classification results are commonly judged using accuracy, precision/recall or F-scores. While accuracy yields misleading results in case of unbalanced classes in the test set, F-scores resolve this issue, but are primarily defined for binary problems with one positive and one negative class. The F-score can be generalized for multiclass problems, e.g. by macro- or micro-averaging [118].

A major disadvantage of classification for pattern recognition is that the availability of reliably labeled data is essential: as labeled instances of process patterns in semiconductor industry need to be generated manually by the expert, they are prone to incompleteness and errors. In addition, pattern types vary by product, i.e. training data must be provided for each product separately. As a result, training data for supervised learning is (at least for some products) not available in a sufficient volume. In deep learning, the concept of transfer learning aims to resolve this issue by transferring trained classifiers from one classification problem to another - however, methods that efficiently deploy this paradigm are still under research.

Another aspect, which has to be considered when applying classification techniques for pattern recognition are new classes: state-of-the-art classifiers are merely able to assign a new instance to one of the known classes, which are present in the training dataset. An intuitive way is to define a residual class of all elements, which are not in the classes present during training. However, this approach suffers from the inhomogeneity of the residual class. Especially if the new samples differ significantly from previously seen instances, this approach is likely to fail.

3.3.3 Semi-supervised Learning

In addition to the classical approaches of supervised and unsupervised learning, the task of integrating new classes into a classifier was investigated in previous work [17]. In general, the idea is to enrich the information from labeled data with additional information from unlabeled data, such that the limited amount of labeled data can be compensated and elements from previously unseen classes can be detected.

The idea builds on the concept of a Bayes classifier, where basically any distribution or covariance structure is possible. In our case, the Gaussian distribution will be used for exemplary purpose. The Bayes classifier is well suited for this application since it is constructed on a generative, statistical foundation, which enables good interpretability of the results - a clear requirement for the investigated use-case.

Related work on classification with new classes Related concepts towards recognizing known as well as unknown classes include one-class and PU-learning (e.g. one-class SVM), open set recognition, exploratory learning, as well as semi-supervised clustering methods, which can be adapted for this purpose.

A similar area, where elements from novel classes can be detected, is open set recognition, introduced by Scheirer et al. [119]. In their work, they aim to distinguish elements from known and unknown classes in a discriminative, supervised setting. For this purpose, they introduce the one-vs-set machine [119], which is then adapted to the W-SVM [120], i.e. the Weibull-calibrated SVM, which outperforms the baseline set by one-class SVMs. While these algorithms are merely used for binary problems (only one known class), extensions of the concept are able to handle multiclass settings. In later works, Scheirer et al. [121] further applied neural network topologies for this problem.

Another related term, exploratory learning, was coined in the PhD dissertation of Dalvi [122] and the related publications, e.g. Dalvi et al. [123]. In the latter work, the authors introduce the problem of classifying instances in the presence of novel, previously unseen classes in a semi-supervised way. Similarly to Scheirer's work, the underlying setup to solve the exploratory learning problem is an SVM, the so-called LACU-SVM, but also versions for other elementary classifiers (e.g. Naive Bayes, etc.) are proposed. A specific focus is put on incremental learning, i.e. integration of new classes over time by a continuing training phase.

The most fundamental ideas regarding the existence of new classes is the one-class SVM. In this concept, the idea is to train an SVM in an one-against-all setting, merely using positive training samples. This method is especially useful, if hardly any instance is available from unknown classes. A similar term is PU-learning [124], i.e. learning from positive and unlabeled data. In contrast to the one-class SVM, the classifier distinguishing known from unknown elements is

trained using positively labeled, as well as unlabeled data, while no instances are labeled from the novel classes. The PU-learning problem is a relaxation of the one-class SVM, which is based on semi-supervised learning.

Finally, unsupervised techniques were adapted in order to fit the semi-supervised classification problem with new classes. For this purpose, clustering methods were combined with constraints, which introduce information obtained from data labels. Examples for this approach are the cGMM (constrained GMM) introduced by Shental et al. [125], as well as the concept of CEC-IB, introduced by Śmieja and Geiger [126]. In general, clustering algorithms are optimized towards different evaluation metrics (e.g. NMI or ASC) compared to classification. Furthermore, randomized permutations of the output categories are possible, since clusters are (in contrast to classes) unordered.

In general, state-of-the-art algorithms for classification in the presence of elements from unknown classes share several limitations, which will be tackled by our suggested method using Bayes classifiers in the following: firstly, most of the mentioned approaches (except for one version of exploratory learning and cGMM) are discriminative, which provides fewer information on the data distribution than a generative approach - yielding lower interpretability. Furthermore, unsupervised methods, such as CEC-IB and cGMM rather differ in their objectives from classification problems. As a third aspect, many state-of-the-art approaches in these related fields (such as the one-class SVM or PU-learning) discriminate between a single known and an unknown class, rather than handling multiple known and unknown classes at the same time.

On the other hand, several approaches (open set recognition, exploratory learning) aim for an automated integration of additional classes to the dataset by incremental learning, which is not an objective for our problem: while integration of new classes can also be implemented with the method proposed in the following, an interference of the expert is essential, since the criticality of the new process pattern needs to be assessed manually, introducing extrinsic information.

A semi-supervised classifier with unknown classes Assuming that a labeled set of wafermaps $\mathbb{L}$ and unlabeled set of wafermaps $\mathbb{D}$ are available, the goal is to train a classifier, which solves a multiclass classification problem and detects elements from previously unknown classes at the same time.

Formally, the labeled training set contains instances from known classes $C = \{c_1, \ldots, c_N\}$. Furthermore, an unlabeled training set consists of elements from $C^+ = C \cup \{c_{N+1}, \ldots, c_{N+k}\}$, where $c_{N+1}, \ldots c_{N+k}$ are unknown classes. The target is to train a classifier f, which assigns each instance in $\mathbb{F}$ to either a class in C, if the instance is from a known class, or to 0, if the instance is from an unknown class. The domain of f is the feature space, while the image C_0 of f is given by $C_0 = C \cup \{0\}$.

In order to solve the problem by Bayesian decision theory, the predicted class c^* is obtained from

$$
\begin{aligned}
c^* &= \arg\min_{c \in C_0} \mathbb{E}_{\tilde{c}|\boldsymbol{x}} V(c, \tilde{c}) \\
&= \arg\min_{c \in C_0} \int_{C^+} V(c, \tilde{c}) \mathrm{d}p(\tilde{c}|\boldsymbol{x}) \\
&\underset{C^+ \text{ countable}}{=} \arg\min_{c \in C_0} \sum_{\tilde{c} \in C^+} V(c, \tilde{c}) \cdot \underbrace{p(\boldsymbol{x}|\tilde{c})}_{\text{likelihood}} \cdot \underbrace{p(\tilde{c})}_{\text{prior}}.
\end{aligned} \tag{3.51}
$$

Here, $V : C_0 \times C^+ \to [0, 1]$ denotes a loss function, which is selected as follows for the classification problem

$$
V(c, \tilde{c}) = \begin{cases} 0 & c = \tilde{c} \\ 0 & c = 0, \tilde{c} \in \{c_{N+1}, \ldots, c_{N+k}\} \\ 1 & \text{else.} \end{cases} \tag{3.52}
$$

The prior probabilities for the known classes in C can be easily obtained by assigning a uniform prior, i.e. $p(c) = \alpha \frac{1}{N}$ for $c \in C$, with a positive parameter α accounting for the previously unknown samples. The parameter α, as well as the prior for the unknown classes $p(0)$ will be specified at a later point.

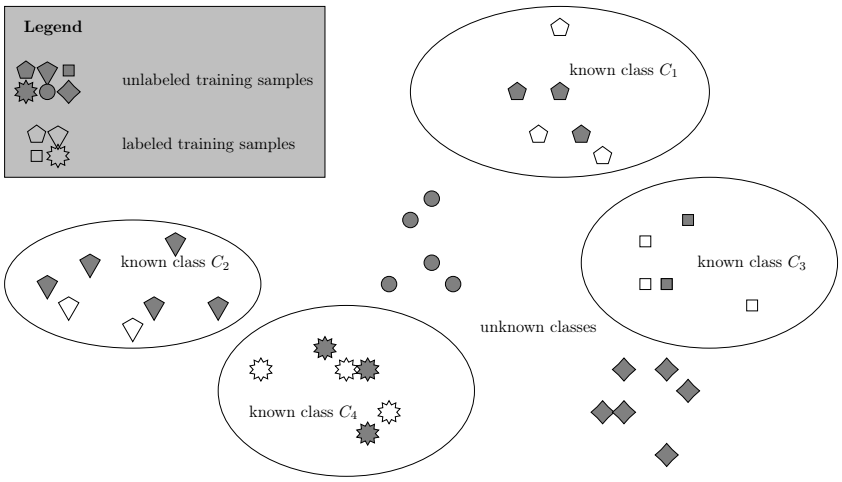
With regard to the likelihood function, a parametric assumption is required (other models are evenly possible): we assume that $\boldsymbol{x}|c \sim \mathcal{N}(\mu^{(c)}, \Sigma^{(c)})$. Optionally, a Bayesian classifier with diagonal structure of the covariance matrix $\Sigma^{(c)}$ (Naive Bayes classifier) can be used, i.e. $p(\boldsymbol{x}|c) = \prod_{i=1}^{d} p(x_i|c)$, $c \in C$. While this model is sufficient for a multiclass classifier on C, we elevate the procedure to classify instances from C^*, i.e. define the prior $p(0)$, as well as the likelihood $p(\boldsymbol{x}|0)$.

The idea to specify likelihood and prior for unknown classes is based on the following 2-step approach, see Fig. 3.9:

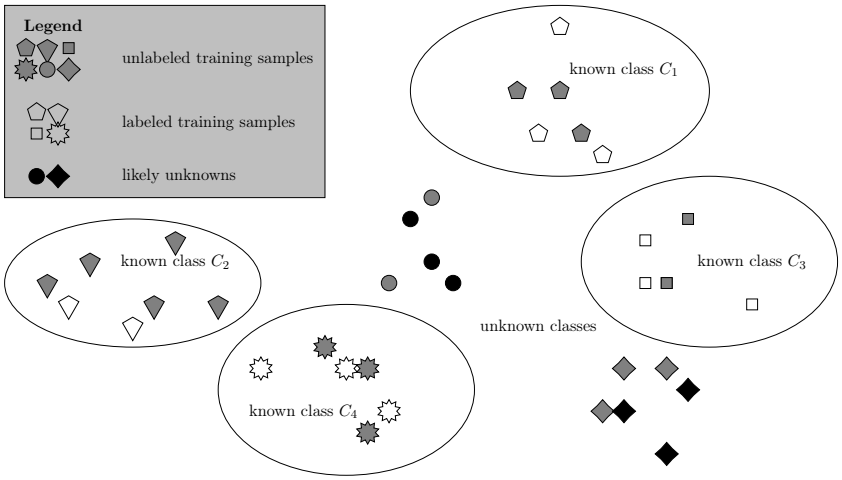
- application of the Spy EM (S-EM) algorithm, presented by Liu et al. [124], to identify elements, which are likely to originate from unknown classes (likely unknowns),
- modeling these likely unknown elements by a Gaussian Mixture Model (GMM) in order to introduce a novel class.

Likelihood and prior for unknown classes The first step, i.e. application of the S-EM algorithm, is used to detect a subset $U \subset \mathbb{D}$ of the unlabeled training data, which contains instances from previously unknown classes. In contrast to the classical application of S-EM presented by Liu et al. [124], multiple known classes are available in our case.

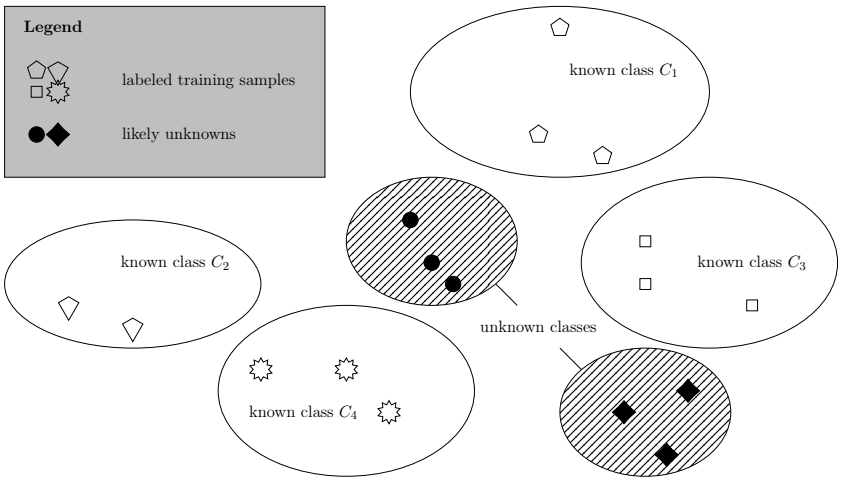
S-EM is applied as follows: given an input parameter $\xi \in [0, 1]$, a subset $S \subset \mathbb{L}$ containing a fraction ξ of the elements in $\mathbb{L}$ is randomly sampled from the labeled training data of all classes in a balanced way. These elements are called spies. When initializing a standard Initial EM (I-EM) algorithm (a semi-supervised version of Expectation Maximization (EM)), all elements from $L \setminus S$ remain with their original labels, while the spies in S and the unlabeled samples in $\mathbb{D}$ are assigned the label 0. I-EM returns a classifier, which approximates the classification w.r.t.



(a) Problem setup of semi-supervised classification with unknown classes.



(b) Step 1: likely unknowns are identified using S-EM.



(c) Step 2: likely unknowns are modeled via a GMM.

Fig. 3.9: Schematic procedure for semi-supervised classification with unknown classes. [17]

C_0. However, since spies from the known classes are labeled with 0, one can calculate a lower limit for the likelihood $q(\boldsymbol{x}|c)$ of class c in the EM-trained Bayes classifier q as follows:

$$t = \min_{\boldsymbol{x} \in S} q(\boldsymbol{x}|c(\boldsymbol{x})), \tag{3.53}$$

given that $c(\boldsymbol{x})$ denotes the correct class of $\boldsymbol{x}$. The set of likely unknowns, i.e. elements, which are not part of the known classes with high probability, is given by

$$U = \{\boldsymbol{x} \in \mathbb{D} : \forall c \in C : \ q(\boldsymbol{x}|c) < t\}. \tag{3.54}$$

In case that U is empty, no unknown class will be introduced. Otherwise, the prior probability for class 0 is calculated from the number of likely unknown element, which are assumed to be a representative sample of the unknown classes: $p(0) = \frac{|U|}{|\mathbb{D}|}$. In order to fulfill the conditions of a probability measure, the parameter α needs to be set to $\alpha = 1 - p(0)$.

The second step of the proposed algorithm, i.e. the estimation of a GMM, serves as a tool to model the conditional likelihood of $(\boldsymbol{x}|0)$. In this case, since we do not know the number of unknown classes k, a mixture model is selected. Due to the choice of Gaussian Bayes classifiers, the mixture model consists of Gaussian distributions (otherwise, the same type of distribution as for the Bayes classifier should be used in the mixture model in order to guarantee that each class is modeled with the same model complexity). The final model for $\boldsymbol{x} \in U$ is obtained as follows:

$$\boldsymbol{x} \sim \sum_{i=1}^{\ell} \phi_i \mathcal{N}(\mu_i, \Sigma_i) =: p(\boldsymbol{x}|0), \tag{3.55}$$

where ℓ is the number of mixture parameters and ϕ_i are the mixture weights, which are, by default, set to $\phi_i = \frac{1}{\ell}$.

Number of mixture components ℓ In general, it will not be reasonable to assume that k can be obtained from the dataset in an unsupervised way due to intra-class variations. However, since the focus is not primarily put on incremental learning, it suffices to estimate a number of mixture parameters ℓ with $\ell \approx k$. We present an adapted version of the Bayesian Information Criterion (BIC), a key number from statistics to judge the model quality [127]. While the common formulation of BIC presents a trade-off between model complexity (judged by the number of parameters) and the number of data points used for estimation, our version considers all parameters (those from the known, as well as from the unknown classes), while varying only the parameters from the unknown classes. Hence, the BIC is obtained as follows:

$$\mathrm{BIC}(\tilde{p}(.;\ell)) = m \log(|U| + |\mathbb{L}|) - \max_{\{\mu_i, \Sigma_i\}} \sum_{\boldsymbol{x} \in U \cup \mathbb{L}} \log(\tilde{p}(\boldsymbol{x}; \ell)). \tag{3.56}$$

The number of estimated model parameters m is calculated by $m = (\ell + N) \cdot d + (\ell + N) \cdot \frac{d(d+1)}{2}$, where the first term includes the number of components estimated for the mean vectors μ_i.

The second term includes the number of components estimated for the non-zero entries of the covariance matrices Σ_i. $\tilde{p}(\boldsymbol{x};\ell)$ denotes the likelihood given known and unknown classes, i.e.

$$\tilde{p}(\boldsymbol{x};\ell) = \frac{1}{N+\ell}\sum_{c\in C} p(\boldsymbol{x}|c) + \frac{\ell}{N+\ell}p(\boldsymbol{x}|0). \tag{3.57}$$

In summary, the number of mixture components ℓ is selected as the positive integer, which minimizes $\mathrm{BIC}(\tilde{p}(.;\ell))$.

Since no additional information is gained about the known classes in C, it holds that $p(\boldsymbol{x}|c) = q(\boldsymbol{x}|c)$ for all $c \in C$. As an overview, Alg. 1 summarizes the steps, which were taken to model the classifier. In this formulation, ℓ_{max} denotes a maximal number of mixture components, which is set to a large positive integer (depending on the complexity of the dataset and computational restrictions).

Algorithm 1 Generative Semi-Supervised Classifier for Unknown Classes

sample $S \subset D$, s.t. $|S| = \xi \cdot |\mathbb{D}|$

$$\tilde{c}(\boldsymbol{x}) \leftarrow \begin{cases} c(\boldsymbol{x}) & \boldsymbol{x} \in \mathbb{L} \setminus S \\ 0 & \boldsymbol{x} \in \mathbb{D} \cup S \end{cases}$$

run EM algorithm with labels $\tilde{c}(\boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{L} \cup \mathbb{D}$, and obtain classifier $q(.|c)$

$t \leftarrow \min_{\boldsymbol{x}\in S} q(\boldsymbol{x}|c(\boldsymbol{x}))$

$U \leftarrow \{\boldsymbol{x} \in \mathbb{D} : \ \forall c \in C : \ q(\boldsymbol{x}|c) < t\}$

for $\ell = 1 : \ell_{\max}$ **do**

  train GMM for U with ℓ components

end for

return $\tilde{p}(.;\ell^*)$, s.t. $\ell^* = \underset{\ell=1:\ell_{\max}}{\arg\min}\ BIC(\tilde{p}(.;\ell))$

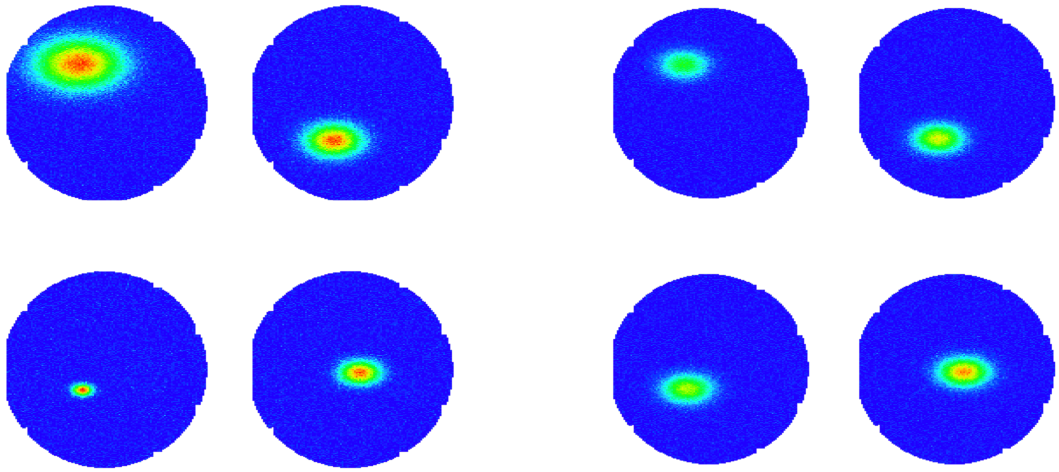
3.4 The Health Factor for Process Patterns

Finally, the Health Factor needs to be defined, based on the information gained from pattern recognition. This concept is presented by Schrunner et al. [18]. Resuming to the formalized problem setup presented in Section 2.4, the aim is to define a loss function V, as well as an appropriate posterior distribution $\mathcal{P}_{\boldsymbol{\theta}|d}$. Apart from the pattern type classification, which was investigated in Section 3.3, two further aspects, intensity and criticality will be introduced and finally combined to the Health Factor.

3.4.1 Intensity Quantification

Methods for intensity quantification are used in order to observe the degree of development of a pattern on the wafer. The concept of quantifying intensity of patterns is rather novel in the field of semiconductor data and was investigated in a Master's thesis accompanying this work [128], as well as presented by Jenul et al. [21].

In general, the intensity $i(\boldsymbol{w}|p)$ of the wafermap $\boldsymbol{w}$ is a value ranging in $[0, 1]$ given the pattern type p, where 0 denotes random noise (no pattern) and 1 denotes a strongly developed process pattern. We assume that intensity is an intrinsic property of a wafermap. Two distinct types of



(a) x-y-intensity (b) z-intensity

Fig. 3.10: Problem setup for intensity quantification [128]. While Fig. 3.10a demonstrates variations of the same pattern type in the x-y-plane (x-y-intensity), Fig. 3.10b depicts variations in the z-plane (z-intensity). All plots are performed on a joint data scale.

intensity are distinguished, demonstrated in Fig. 3.10: x-y- and z-intensity. While x-y-intensity refers to the spatial expansion of the pattern on the wafer, z-intensity relates to the measurement values, i.e. the z-axis. In general, intensity will be specific to a certain pattern type, hence, no universal method matches all pattern types. Instead, distinct methods can be adapted to quantify process pattern intensity on wafermaps.

Autocorrelation methods are one statistical approach to quantify the intensity of process patterns. The autocorrelation $\rho(X_{t_i}, X_{t_j})$ is commonly known for 1-dimensional stochastic processes (e.g. time series), where it describes the correlation of the observed random variable X at different time points t_i and t_j, i.e.

$$\rho(X_{t_i}, X_{t_j}) = \mathbb{E}\left[X_{t_i}, X_{t_j}\right]. \tag{3.58}$$

Moran's I Directly adapting the concept of autocorrelation from 1-dimensional stochastic processes to random fields, we obtain the Moran's I [129], which is calculated as follows:

$$I = \frac{N}{\sum_{i,j} w_{ij}} \frac{\sum_{i,j} w_{ij} \left(X_i - \mathbb{E}\left[X\right]\right)\left(X_j - \mathbb{E}\left[X\right]\right)}{\sum_i \left(X_i - \mathbb{E}\left[X\right]\right)^2}, \tag{3.59}$$

where $\mathbb{E}[X]$ can be replaced with the mean of X and w_{ij} are weights, obtained from a weighting of the neighborhood of X_i. Spatial neighborhoods can be specified in the same way as in Section 3.1.5. A common choice of the weights is to set $w_{ij} = \frac{1}{\text{dist}(X_i, X_j)}$ given a distance metric "dist" with $i \neq j$. Equivalently to the univariate autocorrelation, Moran's I is bounded between -1 and 1, where 1 denotes a clustered pattern, 0 denotes random noise and -1 denotes a dispersed structure.

The intention behind using Moran's I to quantify pattern intensities on wafermaps is that patterns, which hardly deviate from measurement noise will be assigned a Moran's I value $I \approx 0$. However, if a strong pattern is present, coherent regions are likely to be present on the wafermap, which implies that the Moran's I value will increase, i.e. $I \gg 0$. Dispersed patterns, i.e. Moran's I values below 0 are not likely due to their optical structure and will not be considered for intensity quantification.

Spatial entropy Another concept, which can be transferred to the spatial case is Shannon entropy [130]. Shannon entropy is a measure for the information content delivered by a probability distribution. Shannon entropy is defined as

$$H(X) = \mathbb{E}\left[-\log_2(X)\right] = \int\limits_{x \in \mathrm{supp}_X} \log_2\left(\frac{1}{\mathcal{P}(x)}\right) d\mathcal{P}(x), \tag{3.60}$$

where $\mathcal{P}$ is the probability mass function of X and supp_X is the support of X. While this formula does not take spatial structures into account, the Spatial entropy [131] extends the concept by binning all data points of the domain into equidistant bins $\{1, \ldots, r\}$ w.r.t. their measurement values and adding intra distance d_i^{int} and extra distance d_i^{ext} for each bin i:

$$H_s(X) = \sum_{i=1}^{r} \frac{d_i^{int}}{d_i^{ext}} \mathcal{P}(x_i) \log_2\left(\frac{1}{\mathcal{P}(x_i)}\right). \tag{3.61}$$

In this case, the binning is used as a discretization of the probability distribution of measurement values over the wafermap, i.e. $\mathcal{P}(x_i)$ is estimated by the relative frequency of devices assigned to bin i. The intra and extra distances are calculated within or between each bin:

$$d_i^{int} = \begin{cases} \frac{1}{|X_i| \cdot (|X_i|-1)} \sum\limits_{j \in X_i} \sum\limits_{k \in X_i, k \neq j} \mathrm{dist}(j,k) & \text{if} |X_i| > 1, \\ \lambda & \text{otherwise}, \end{cases} \tag{3.62}$$

$$d_i^{ext} = \begin{cases} \frac{1}{|X_i| \cdot |X \setminus X_i|} \sum\limits_{j \in X_i} \sum\limits_{k \in X \setminus X_i} \mathrm{dist}(j,k) & \text{if } r \neq 1, \\ \beta & \text{otherwise}, \end{cases} \tag{3.63}$$

for a small constant λ and a large constant β.

In contrast to Moran's I, spatial entropy yields low values, if a pattern occurs, i.e., if the information gain is low, which is the case, where a lot of information, indicated by a strong pattern, is already present. High entropy values indicate that few information is present in the data, due to very unspecific (weak) patterns.

More complex approaches towards pattern intensity quantification are presented by Jenul [128], e.g. using Kriging methods or models based on Spatial Point Patterns. However, such models mostly require well-founded knowledge on the specific pattern type, which cannot be provided in an automated way for real-world wafermaps.

Image segmentation If further patterns are present on the wafermap in addition to the pattern of interest, provoked e.g. by test patterns, the intensity quantification measure might fail. Such

relicts are referred to as background structures. Although Assumption 2 (Section 2.3.1) should guarantee the uniqueness of each pattern, such mixtures can occasionally occur and distort the intensity quantification step.

In order to resolve this problem, the presented autocorrelation methods can be combined with image segmentation: using e.g. the k-means algorithm, the ROIs are detected and the quantifier is restricted to these regions. In order to discriminate the regions of interest from background, the median measurement value is calculated among each region obtained from image segmentation and compared to the global median on the wafermap. The regions of interest are defined as those with the largest absolute difference from the global median.

Restricting the wafer intensity quantifiers to specific ROI enhances the robustness of the intensity quantification methods w.r.t. perturbations by background structures. However, k-means clustering yields non-deterministic results and, as all image segmentation techniques, must be calculated on each wafermap separately, leading to higher computational effort. Alternative image segmentation methods, such as MRF- or CNN-based methods, would be applicable, but processing times will probably increase significantly using these approaches.

Feature selection & machine learning Unfortunately, it is not possible to cover all types of process patterns with a single intensity quantifier - this results from the large diversity of pattern types. With the presented methods, as well as with more trivial approaches (e.g. quantifying intensity by the relative size of the ROI, compared to the total size of the wafermap), a set of promising features can be collected. In order to decide, which feature (or feature combination) fits the instances of a specific pattern type best, unsupervised machine learning methods can be applied. For this purpose, Jenul [128] investigated two possibilities:

- PCA: calculating the most promising linear combination of features by applying PCA and defining the first principal component as the intensity quantifier (i.e. the principal component, which covers the largest portion of the total variability in the data),
- unsupervised feature selection: applying a feature selection criterion to decide for one specific feature, which covers most information, based on Shannon entropy.

In general, the feature selection approach yields more interpretable results, since only a single feature is selected instead of a linear combination. However, in more complex cases, where the intensity cannot be completely covered by a single feature, PCA might be more appropriate. On the other hand, PCA might revert the scale of the intensity quantifier by assigning negative weights to the features. Both approaches are valid and do not require manual inputs or intensity labels, but can rather be deployed on the training samples for each pattern type, which are used for the pattern type classifier. Hence, detecting a suitable combination or selecting a suitable feature does not request any additional input from the expert.

3.4.2 Pattern Criticality

Another aspect that must be taken into account is pattern criticality, which refers to the judgment of the expert whether process patterns indicate critical or uncritical deviations, based on domain knowledge and historical data. Since expert knowledge cannot be provided for each wafermap during training, we define pattern criticality at the aggregation level of pattern types, i.e. the expert assigns a criticality level to each pattern type, which is specified during training.

Implicitly, this implements the underlying assumption that a link exists between the type of process pattern present on the wafermap and a specific kind of deviation in the production process. The expert judges the severity of the process deviation by assigning a value to the criticality of the pattern type. In contrast to the intensity concept, the criticality value is an extrinsic information, which is not covered in the wafer test data.

The criticality $h(p)$ maps the pattern type p to a value in $[0, 1]$, assigning 0 to uncritical process patterns and 1 to critical ones. Since in this study, no automated method is applied for this step, the granularity of values is left to the expert, i.e. a simple assignment could be to select $h(p) \in \{0, 1\}$ for each pattern type. However, if more information is available for a product and more critical process patterns exist, it is reasonable to use a refined categorization, e.g. $h(p) \in \left\{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\right\}$ - the resulting Health Factor is able to include this weighting to return a more specific indication of the risk level.

While a manual selection of the criticality for certain pattern types might be useful (e.g., if the expert is interested in identifying and recognizing a specific pattern type out of all others), automatized and semi-automatized heuristics to select the criticality level from additional data sources are conceivable, such as

- setting the criticality in relation to the yield loss, which occurred for the training data showing the concerned pattern type,
- setting the criticality to a (normalized) cost function, describing the expected yield loss compared to the costs of maintenance to resolve the specific process deviation,
- obtaining the criticality from other automated tools, which e.g. judge correlations between patterns and deviations from inline process data,
- setting the criticality to 1 for each pattern in order to reduce the Health Factor to purely judge the pattern type-specific intensity levels.

Further heuristics to assist the expert might be available from more specific applications. In general, criticality is a parameter, which enables the expert to control the intended output of the Health Factor - its manual selection raises many possibilities to transfer the concept of the Health Factor from process monitoring to new areas, such as e.g. quality control.

3.4.3 Mathematical Concept

As presented in recent work [18], it is possible to combine the three components, pattern type, pattern intensity and pattern criticality in an adequate way. Therefore, a loss function V will be specified, first. By definition, the loss is a deterministic function, assigning a value in $[0, 1]$ to each possible state of nature $\boldsymbol{\theta} \in \Theta$. In the first step, it will be assumed that the pattern type is a probabilistic measure, while intensity and criticality are deterministic. In the following, this assumption will be weakened, providing versions for probabilistic criticality and intensity, as well.

Deterministic intensity and criticality Since the uncertainty is merely covered by the pattern type $p \in \mathbb{P}$, the states of nature are given by the possible pattern types, i.e. $\Theta = \mathbb{P}$. In case that a specific state of nature θ occurs, the loss function $V(\theta)$ denotes the effect, which results from this circumstance. Given a specific pattern type on a wafermap, this effect is determined

by the criticality of the pattern type on the one hand (i.e. the more critical this pattern type is, the higher should the Health Factor be) and by the intensity on the other hand (i.e. a higher pattern intensity should imply a higher Health Factor). For both components, the criticality as well as the intensity range in $[0, 1]$, the most intuitive solution is to determine the loss function V by

$$V(\boldsymbol{w}; p) = h(p) \cdot i(\boldsymbol{w}; p). \tag{3.64}$$

Hence, the Bayesian posterior expected loss $\mathcal{R}$, which defines the final Health Factor $HF_{\boldsymbol{w}}$, is determined as follows:

$$\begin{aligned} HF_{\boldsymbol{w}} &= \int_{\theta \in \Theta} V(d, \theta)\ d\mathcal{P}_{\theta|d}(\theta) \\ &= \sum_{p \in \mathbb{P}} h(p) \cdot i(\boldsymbol{w}; p) \cdot \mathcal{P}_{p|\boldsymbol{w}}(p). \end{aligned} \tag{3.65}$$

At this stage, different options are possible to specify the posterior probability $\mathcal{P}_{p|\boldsymbol{w}}(p)$ from the pattern type classifier. The predicted class $p^*(\boldsymbol{w}) \in \mathbb{P}$ for an instance $\boldsymbol{w}$ can be inserted as a binary measure (hard classification):

$$\mathcal{P}_{p|\boldsymbol{w}}(p) = \begin{cases} 1 & p = p^*(\boldsymbol{w}) \\ 0 & \text{else.} \end{cases} \tag{3.66}$$

If the classifier delivers a discrete probability distribution $\mathcal{P}(p|\boldsymbol{w}) = \left\{\tilde{p}_1(\boldsymbol{w}), \ldots, \tilde{p}_{|\mathbb{P}|}(\boldsymbol{w})\right\}$ given a wafermap $\boldsymbol{w}$, soft classification can be exploited by setting $\mathcal{P}(.|\boldsymbol{w}) = \mathcal{P}(\boldsymbol{w})$. The advantage of applying hard classification is that a more specific result is obtained, on the other hand uncertainties are represented by soft classification in a more adequate way.

If a generative classifier is used, the classification approach can be decomposed into likelihood $L(\boldsymbol{w}; p) = \mathcal{P}_{\boldsymbol{w}|p}(\boldsymbol{w})$ and prior distribution $\mathcal{P}_p(p)$. In this case, (3.65) reads as follows:

$$HF_{\boldsymbol{w}} = \sum_{p \in \mathbb{P}} h(p) \cdot i(\boldsymbol{w}; p) \cdot \mathcal{P}_{\boldsymbol{w}|p}(\boldsymbol{w}) \cdot \mathcal{P}_p(p). \tag{3.67}$$

Probabilistic intensity and deterministic criticality In case that uncertainty arising from the intensity quantifier shall be introduced, the Health Factor can be adapted. In particular, the state of nature is 2-dimensional and comprises the information on the pattern type $p \in \mathbb{P}$ and

the intensity $i \in [0,1]$, i.e. $\Theta = \mathbb{P} \times [0,1]$. Thus, the loss function reads $V(\boldsymbol{w}; p, i) = h(p) \cdot i$, which leads to the following expression for $HF_{\boldsymbol{w}}$:

$$\begin{aligned} HF_{\boldsymbol{w}} &= \int_{\boldsymbol{\theta} \in \Theta} V(d, \boldsymbol{\theta}) \; d\mathcal{P}_{\boldsymbol{\theta}|d}(\boldsymbol{\theta}) \\ &= \int_{(p,i) \in \mathbb{P} \times [0,1]} V(\boldsymbol{w}; p, i) \; d\mathcal{P}_{(p,i)|\boldsymbol{w}}(p, i) \\ &= \sum_{p \in \mathbb{P}} \int_0^1 V(\boldsymbol{w}; p, i) \cdot \mathcal{P}_{i|p,\boldsymbol{w}}(i) \cdot \mathcal{P}_{p|\boldsymbol{w}}(p) \; di \\ &= \sum_{p \in \mathbb{P}} \int_0^1 h(p) \cdot i \cdot \mathcal{P}_{i|p,\boldsymbol{w}}(i) \; di \cdot \mathcal{P}_{p|\boldsymbol{w}}(p) \\ &= \sum_{p \in \mathbb{P}} h(p) \cdot \mathbb{E}\left[i|p, \boldsymbol{w}\right] \cdot \mathcal{P}_{p|\boldsymbol{w}}(p). \end{aligned} \tag{3.68}$$

As a result, no information on the distribution of the intensity i given a pattern type p and a wafermap $\boldsymbol{w}$ is required, apart from the expected value $\mathbb{E}\left[i|p, \boldsymbol{w}\right]$ of this conditional distribution. Furthermore, the structure of the explicit representation of the Health Factor is the same to the purely deterministic case.

Probabilistic intensity and criticality Finally, the option that both, intensity i and criticality h are assumed to be random is investigated. Hence, the state of nature contains all three components, i.e. $\Theta = \mathbb{P} \times [0,1]^2$. The according loss function is obtained by $V(\boldsymbol{w}; p, i, h) = h \cdot i$. This version is reasonable in case that, e.g., the criticality value h is selected from an external data-driven procedure as suggested in Section 3.4.2, where information on the distribution is delivered instead of a point estimate.

In the investigated case, an additional assumption on the probabilistic dependencies between the single components is necessary: since both, intensity i and criticality h are random variables, but obtained from distinct sources (i is intrinsically obtained from the wafermap, while h comprises external knowledge), it can be assumed that no additional knowledge about h is present in i. Furthermore, pattern criticality is assumed to be independent from the wafermap given the pattern type, i.e. the criticality level is specified for each pattern type rather than for each wafermap. In summary, these criteria imply that the following assumption holds:

$$\mathcal{P}_{h|p,\boldsymbol{w},i}(h) = \mathcal{P}_{h|p}(h). \tag{3.69}$$

The Health Factor can be obtained in a similar way as in the deterministic version as follows:

$$
\begin{aligned}
HF_{\boldsymbol{w}} &= \int_{\boldsymbol{\theta}\in\Theta} V(d,\boldsymbol{\theta})\; d\mathcal{P}_{\boldsymbol{\theta}|d}(\boldsymbol{\theta}) \\
&= \int_{(p,i,h)\in\mathbb{P}\times[0,1]^2} V(\boldsymbol{w};p,i,h)\; d\mathcal{P}_{(p,i,h)|\boldsymbol{w}}(p,i,h) \\
&= \sum_{p\in\mathbb{P}} \int_0^1 \int_0^1 V(\boldsymbol{w};i,p,h)\cdot \underbrace{\mathcal{P}_{h|p,\boldsymbol{w},i}(h)}_{(3.69)} \cdot \mathcal{P}_{i|p,\boldsymbol{w}}(i)\cdot \mathcal{P}_{p|\boldsymbol{w}}(p)\; di\; dh \\
&= \sum_{p\in\mathbb{P}} \int_0^1 \int_0^1 h\cdot i\cdot \mathcal{P}_{h|p}(h)\cdot \mathcal{P}_{i|p,\boldsymbol{w}}(i)\cdot \mathcal{P}_{p|\boldsymbol{w}}(p)\; di\; dh \\
&= \sum_{p\in\mathbb{P}} \int_0^1 h\cdot \mathcal{P}_{h|p}(h)\; dh \cdot \int_0^1 i\cdot \mathcal{P}_{i|p,\boldsymbol{w}}(i)\; di \cdot \mathcal{P}_{p|\boldsymbol{w}}(p) \\
&= \sum_{p\in\mathbb{P}} \mathbb{E}\left[h|p\right]\cdot \mathbb{E}\left[i|p,\boldsymbol{w}\right]\cdot \mathcal{P}_{p|\boldsymbol{w}}(p). \qquad (3.70)
\end{aligned}
$$

As a result, the expression is, again, similar to the previous versions, replacing the deterministic term for criticality by the expected value. Thus, the required information from the distribution of the criticality parameter, as well as the information required from the intensity distribution is covered by their first moments.

3.4.4 Elevating the Aggregation Level: The Health Factor for Wafers and Lots

In semiconductor production, decisions are mainly made on higher aggregation levels, since investigating single wafermaps requires large effort. Hence, information on process and product quality is commonly aggregated to wafers or lots.

In order to compare the existing methods with the Health Factor, an extension of the concept to wafer or lot level must be provided. For elevating the aggregation level, it is necessary to guarantee that no critical wafermap is neglected, i.e. the Health Factor of the wafer / lot has to be at least at the level of the highest Health Factor on wafermap level. The main question that arises, is whether two or more wafermaps depicting critical process patterns should cumulate in the wafer-level Health Factor. It is known from process experts that equal patterns tend to be present in multiple wafermaps of the same wafer, if they originate from related test parameters. However, since these wafermaps show the same process deviation, a cumulation of the Health Factor values would result in judging the test sequence rather than the process quality, i.e. if one parameter showing a critical pattern is measured more often, the Health Factor increases although the process deviation is the same in both cases. Hence, the wafer-Health Factor HF_W, where a wafer is defined by a set of wafermaps $W = \{\boldsymbol{w}_i, i = 1,\ldots,i_{\max}\}$, is defined as

$$
HF_W = \max_{\boldsymbol{w}\in W} HF_{\boldsymbol{w}}. \qquad (3.71)
$$

	W_1	W_2	W_3	...
p_1	$HF_{W_1}^{p_1}$	$HF_{W_2}^{p_1}$	$HF_{W_3}^{p_1}$	...
p_2	$HF_{W_1}^{p_2}$	$HF_{W_2}^{p_2}$	$HF_{W_3}^{p_2}$	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$
p_L	$HF_{W_1}^{p_L}$	$HF_{W_2}^{p_L}$	$HF_{W_3}^{p_L}$	...
overall	HF_{W_1}	HF_{W_2}	HF_{W_3}	...

Tab. 3.5: Demonstration of an evaluation matrix, depicting the Health Factor on wafer level (wafer-Health Factor), as well as the pattern-Health Factor values for each pattern $p \in \mathbb{P} = \{p_1, \ldots, p_L\}$. The wafer-Health Factor values (overall) equals the column maxima.

The same arguments are valid when extending the concept to lots: a lot-Health Factor must not go below the highest wafer-Health Factor, on the other hand, a cumulation does not yield any benefits (although, in this case, it is independent from the test sequence). Defining a lot L as an ensemble of wafers $L = \{W_i, i = 1, \ldots, \hat{i}_{\max}\}$, the lot-Health Factor is denoted as follows:

$$HF_L = \max_{W \in L} HF_W = \max_{w \in W \in L} HF_{\boldsymbol{w}}. \tag{3.72}$$

Another version of the Health Factor, which provides relevant information for the expert is the pattern-Health Factor, pursuing the following intention: since the Health Factor cumulates knowledge about all pattern categories, the expert might be specifically interested in receiving an overview on the most critical pattern types contributing to the indicator on one wafer / in one lot. Hence, the pattern-Health Factor is a refinement of the Health Factor, which summarizes the highest Health Factor values obtained from each pattern type. Given that $p \in \mathbb{P}$ is a pattern type, which was used for training, the pattern-Health Factor HF_W^p for a wafer W is defined as

$$HF_W^p = \max\{HF_{\boldsymbol{w}^*} : \boldsymbol{w}^* = \arg\max_{\boldsymbol{w} \in W} \mathcal{P}_{p|\boldsymbol{w}}(p)\}. \tag{3.73}$$

Since it holds that $HF_W = \max_{p \in \mathbb{P}} HF_W^p$, the pattern-Health Factor can be used to instantaneously receive information on the type of pattern, which is responsible for a high wafer-Health Factor. The same principle is transferable to the lot-Health Factor. Tab. 3.5 demonstrates how the different Health Factor definitions can be combined to an evaluation matrix.

Since different versions of the Health Factor were stated and presented above, Health Factor refers to the version based on wafermaps in the evaluation section (Section 4), if not explicitly stated otherwise.

3.4.5 Consistency Property

The proposed Health Factor is based on the well-established approach of Bayesian statistical decision theory - however, it is not yet clear whether the indicator fulfills the relevant properties for being integrated into an industry 4.0 environment. Unfortunately, abstract formulations of necessary conditions for real-world decision support systems, which need to be satisfied to guarantee consistency, are hardly available in literature. Hence, such a consistency condition for

process monitoring indicators using machine learning is proposed in Schrunner et al. [19] - the general framework is compatible with various data types and problem setups in process control.

For this purpose, we define the setup for an indicator $\psi : \mathcal{D} \to \mathbb{R}$, i.e. a key value, which is used to track the stability of the process based on information from the data space $\mathcal{D}$. In order to specify the conditions, we require that $(\mathcal{D}, \text{dist})$ is at least a pseudometric space, i.e. $\text{dist} : \mathcal{D} \times \mathcal{D} \to \mathbb{R}^+$ is a distance measure, which fulfills the conditions of non-negativity, symmetry, as well as the triangle equation, see Section 3.3.1, but may violate the definiteness property.

On the data space, we assume that a subset of the data space $\Lambda \subset \mathcal{D}$ was observed previously, e.g. in historical data. For these known instances, expert knowledge $\mathcal{E}$ is available, i.e. the expert can judge whether these process states were critical or not. Expert knowledge is represented by a value between 0 and 1, which can be assigned to each instance in Λ, i.e. $\mathcal{E} : \Lambda \to [0, 1]$. Since the indicator should be able to integrate this historical knowledge, we denote it by $\psi_{\Lambda,\mathcal{E}}$ in the following.

Consistency condition One intuitive condition, which is necessary to be applicable in practice is that the indicator returns positive values, i.e. $\psi_{\Lambda,\mathcal{E}} > 0$, assuming that high values indicate critical states of the process. Hence, thresholding the value is possible in case that e.g. an action or alarm should be triggered by monitoring values above the threshold. Ideally (but not necessarily), an upper bound of the values can also be specified, e.g. $\psi_{\Lambda,\mathcal{E}} \in [0, 1]$.

However, a more restrictive condition must be specified in order to guarantee that the indicator fits to the values specified by the expert knowledge: intuitively, a justified requirement is that the indicator values are similar to the expert judgement, if the observed data is similar to a data point labeled by the expert. In mathematical terms, this *consistency* condition can be expressed as follows: for all $\boldsymbol{d} \in \Lambda$ and $\boldsymbol{x} \in \mathcal{D}$, it holds that

$$|\psi_{\Lambda,\mathcal{E}}(\boldsymbol{x}) - \mathcal{E}(\boldsymbol{d})| \leq \alpha \ \text{dist}(\boldsymbol{x}, \boldsymbol{d}) + \beta, \tag{3.74}$$

where α, $\beta > 0$, β is a small scalar and $\text{dist}(\boldsymbol{x}, \boldsymbol{d}) \leq \text{dist}_{\max}$ for a constant $\text{dist}_{\max} > 0$. Hence, we can conclude that $\psi_{\Lambda,\mathcal{E}}(\boldsymbol{x}) \approx \mathcal{E}(\boldsymbol{d})$ for $\text{dist}(\boldsymbol{x}, \boldsymbol{d}) \to 0$.

In a probabilistic setting, where uncertainty is taken into account for the decision, it is necessary to relax the stated condition to the *relaxed consistency*, i.e.

$$\mathcal{P}(|\psi_{\Lambda,\mathcal{E}}(\boldsymbol{x}) - \mathcal{E}(\boldsymbol{d})| \leq \alpha \ \text{dist}(\boldsymbol{x}, \boldsymbol{d}) + \beta) \geq \xi, \tag{3.75}$$

where $\mathcal{P}$ is the associated probability measure and $\xi > 0$ is a high probability value (slightly lower than 1). In other words, the consistency condition is fulfilled at least with a high probability.

As a special case of the stated condition, we obtain that applying the indicator to a data point $\boldsymbol{d} \in \Lambda$, which was already observed historically, yields a similar value as specified by the expert knowledge $\mathcal{E}$, i.e.

$$\psi_{\Lambda,\mathcal{E}}(\boldsymbol{d}) = \mathcal{E}(\boldsymbol{d}) \pm \beta. \tag{3.76}$$

In case that extrinsic and intrinsic properties have an effect on the indicator, the concept of intensity can be integrated into the framework: we assume that an intensity function I with

$I(\boldsymbol{x}) \geq 0$ for all $\boldsymbol{x} \in \mathcal{D}$ quantifies such influences. Hence, the generalized indicator $\psi_{\Lambda,\mathcal{E},I}$ can be defined via a functional φ, which combines the knowledge from the expert and the historical data with the intrinsic properties, i.e. $\psi_{\Lambda,\mathcal{E},I} = \varphi(\psi_{\Lambda,\mathcal{E}}, I)$. A default choice is $\varphi(f_1, f_2)(\boldsymbol{x}) = f_1(\boldsymbol{x}) \cdot f_2(\boldsymbol{x})$. For this generalized setting, the consistency condition can be expressed as follows:

$$|\psi_{\Lambda,\mathcal{E},I}(\boldsymbol{x}) - \varphi(\mathcal{E}(\boldsymbol{d}), I(\boldsymbol{d}))| \leq \alpha_1 \text{ dist}(\boldsymbol{x}, \boldsymbol{d}) + \alpha_2 |I(\boldsymbol{x}) - I(\boldsymbol{d})| + \beta, \tag{3.77}$$

where $\alpha_1, \alpha_2 > 0$. By setting $I \equiv c > 0$, the original setting without the intensity concept can be obtained as a special case.

The Health Factor as a special case of the indicator framework According to its definition based on statistical decision theory, the Health Factor fulfills the relaxed consistency condition - however, note that we assume a hard classification version of the Health Factor here. In detail, we assume that in the pattern type classifier, the resulting probability measure $\mathcal{P}_{p|\boldsymbol{x}}$ is of binary type. The pattern type, which is assigned with probability 1 to the wafermap, is denoted as $c^*(.)$. In contrast, we denote the ground truth class label by $c(.)$.

A pseudometric can be obtained from the setting of supervised classification problems, first. As shown in [17], a pseudometric is induced by a supervised classifier c^* via the definition

$$\text{dist}(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} 0 & c^*(\boldsymbol{x}) = c^*(\boldsymbol{y}) \\ 1 & \text{else.} \end{cases} \tag{3.78}$$

The data space $\mathcal{D}$ corresponds to the feature space $\mathbb{F}$. In general, the non-negativity assumption of the indicator is trivially fulfilled by the Health Factor. Note that for the Health Factor, the information of criticality is specified per pattern type rather than for single training samples - hence, it holds that $\psi_{\Lambda,\mathcal{E}}(\boldsymbol{x}) = \psi_{\Lambda,\mathcal{E}}(\boldsymbol{y}) = h(c^*(\boldsymbol{x}))$ for $c^*(\boldsymbol{x}) = c^*(\boldsymbol{y})$. It results from the definition of the classifier, which is trained to approximate the ground truth classes $c(\boldsymbol{x})$ by $c^*(\boldsymbol{x})$ that $\mathcal{P}(c^*(\boldsymbol{x}) = c(\boldsymbol{x}))$ is almost 1, hence it holds with high probability that

$$\psi_{\Lambda,\mathcal{E}}(\boldsymbol{d}) = h(c^*(\boldsymbol{d})) = h(c(\boldsymbol{d})) = \mathcal{E}(\boldsymbol{d}) \; \forall \boldsymbol{d} \in \Lambda. \tag{3.79}$$

Finally, we consider for two elements $\boldsymbol{x} \in \mathcal{D}$ and $\boldsymbol{d} \in \Lambda$ in a local neighborhood, such that $\text{dist}(\boldsymbol{x}, \boldsymbol{d}) < 1$. Since the distance measure dist delivers a value $\text{dist}(\boldsymbol{x}, \boldsymbol{d}) < 1$ only in case that $c^*(\boldsymbol{x}) = c^*(\boldsymbol{d})$, $\boldsymbol{x}$ and $\boldsymbol{d}$ are assigned to the same pattern type. Hence, it holds that

$$\begin{aligned} |\psi_{\Lambda,\mathcal{E},i}(\boldsymbol{x}) - \varphi(\mathcal{E}(\boldsymbol{d}), I(\boldsymbol{d}))| &= |\underbrace{h(c^*(\boldsymbol{x}))}_{=h(c^*(\boldsymbol{d}))} \cdot i(\boldsymbol{x}) - h(c^*(\boldsymbol{d})) \cdot i(\boldsymbol{d})| \\ &\underset{(3.79)}{=} |h(c(\boldsymbol{d})) \cdot i(\boldsymbol{x}) - h(c(\boldsymbol{d})) \cdot i(\boldsymbol{d})| \\ &= |h(c(\boldsymbol{d}))| \cdot |i(\boldsymbol{x}) - i(\boldsymbol{d})|, \end{aligned} \tag{3.80}$$

and hence the condition is trivially fulfilled. More details on the flexibility of the indicator framework for process monitoring in industry 4.0 are provided in Schrunner et al. [17].

4 Experiments

4.1 Concept of the Experiments

The concept of the Health Factor, which was presented and explained in the preceding chapters, is sound from the viewpoint of mathematics and fulfills major consistency requirements. Still, the practical value of the method, which is of particular interest from the industrial perspective, needs to be demonstrated. Hence, experiments using simulated and real-world data will be carried out in the following.

The experiments are intended to provide evidence for two distinct aspects: evaluation of the single Health Factor components (in particular pattern type classification, see Section 4.3 and pattern intensity quantification, see Section 4.4), as well as validation of the overall Health Factor system (Section 4.5).

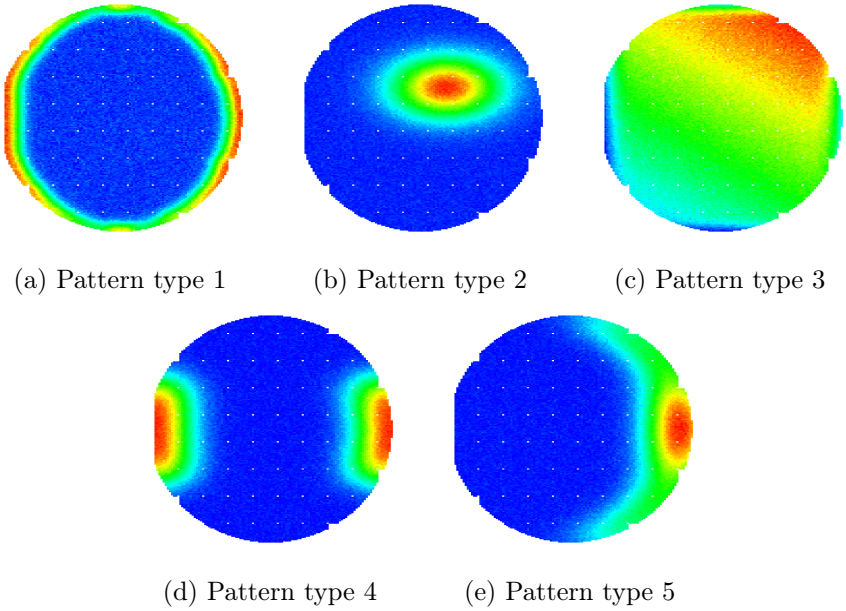
The evaluation of the data-driven Health Factor components, pattern type and pattern intensity, is required to guarantee an accurate performance of these single parts. A particular focus will be set on the pattern type classifier, since a bad performance of this core element seriously distorts the whole system. Furthermore, the pattern intensity will be evaluated - however, the number of datasets, where intensity values can be assessed in comparison to an accurate "ground truth" are limited. Pattern criticality is handled as an extrinsic aspect based on expert judgment and hence will not be evaluated independently.

The validation of the Health Factor concept as a whole gives insight into the benefits and limitations of the system. In this part, the application of the system in practical situations is demonstrated. In general, no ground truth value is available for comparison, since the Health Factor is defined as a novel monitoring parameter, i.e. it covers aspects, which were not available previously. Nevertheless, in order to validate that the results are useful in reality, a binary setup is chosen, where the thresholded Health Factor parameter is used to identify wafermaps with critical patterns at a high level of development. The applied threshold is varied to demonstrate the behavior of the system.

In Section 4.6, we will discuss and interpret the results and draw higher-level conclusions. The strengths and weaknesses will be outlined, especially focussing on both, theoretical and practical aspects. Going beyond the information obtained from the evaluation measures, additional aspects such as runtime or interpretability are analyzed.

4.2 Datasets

The experiments are based on different wafer test datasets, which originate from both, simulation as well as real-world manufacturing. In particular, one simulated dataset (dataset 1) and two



(a) Pattern type 1 (b) Pattern type 2 (c) Pattern type 3

(d) Pattern type 4 (e) Pattern type 5

Fig. 4.1: Pattern types of dataset 1, generated from the simulation procedure in Alg. 2.

distinct real-world datasets (dataset 2 and dataset 3) will be used. For all real-world wafer test data the product, lot and test names are anonymized due to confidentiality reasons, as well as the data scales of the single wafermaps are removed.

Unfortunately, wafer test data are hardly available as open source datasets, thus the possibilities for comparing to state-of-the-art approaches are limited. The largest wafer test dataset for machine learning, which is currently available for public use, is the WM-811K [59] dataset. This dataset shows wafermaps depicting one out of 9 pattern types, including a class for no patterns at all. However, the dataset consists of pass/fail data, hence it is not possible to evaluate the suggested Health Factor on this dataset.

In order to support the development towards reproducible research for data science in industrial applications, we provided the utilized simulated dataset as an open source resource, publicly available on the ZENODO platform, see Pleschberger et al. [77].

4.2.1 Simulated Wafer Test Data

In general, the simulation of wafermaps depicting process patterns can be performed at an arbitrary level of complexity. However, since the variety of characteristics is extremely broad in real-world data, it is almost impossible to cover all eventualities in a simulated dataset. Instead, we aim to simulate wafer test data, showing a diverse subset of frequently occurring pattern types with distinct properties to prove the suggested concept.

The simulation algorithm is initiated with a wafermap template, i.e. a map of spatial x-y-coordinates, which indicates the positions on the wafer, where a device is processed. The template can be obtained from a real-world wafermap by extracting the metadata columns from the dataset, see Tab. 2.1.

With the procedure presented in the following, it is possible to generate 5 distinct pattern types. These 5 types are depicted in Fig. 4.1 for exemplary purpose. The choice of the pattern types

is based on real-world situations and coordinated with product experts. As outlined in the following, the 5 pattern types show individual characteristics, i.e.

- Pattern 1: a ring at the border of the wafermap. The ring varies in its size, but is homogeneously present on the whole wafer border.
- Pattern 2: a spot or clustered region on the wafermap. The spot varies w.r.t. position and size.
- Pattern 3: a homogeneous gradient on the wafermap. The gradient rotates, selecting one of the 8 main directions (horizontal, vertical, diagonal).
- Pattern 4: two spots at opposite positions of the wafer. The spots vary in their sizes.
- Pattern 5: a crescent-shaped region on the border of the wafer. The crescent, again, varies in its size.

In addition, the scales are adapted to an arbitrary range for all wafermaps. As a result, strong and weak samples of the distinct pattern types are generated.

Algorithm 2 Simulation of wafer test data

- initialize the template of the wafermap with 0 at each device position
- set pattern-characteristic positions to 1
- apply a Gaussian blur filter convolution
- transform to an arbitrary data range
- add Gaussian white noise
- introduce outliers at random positions, as well as NAs

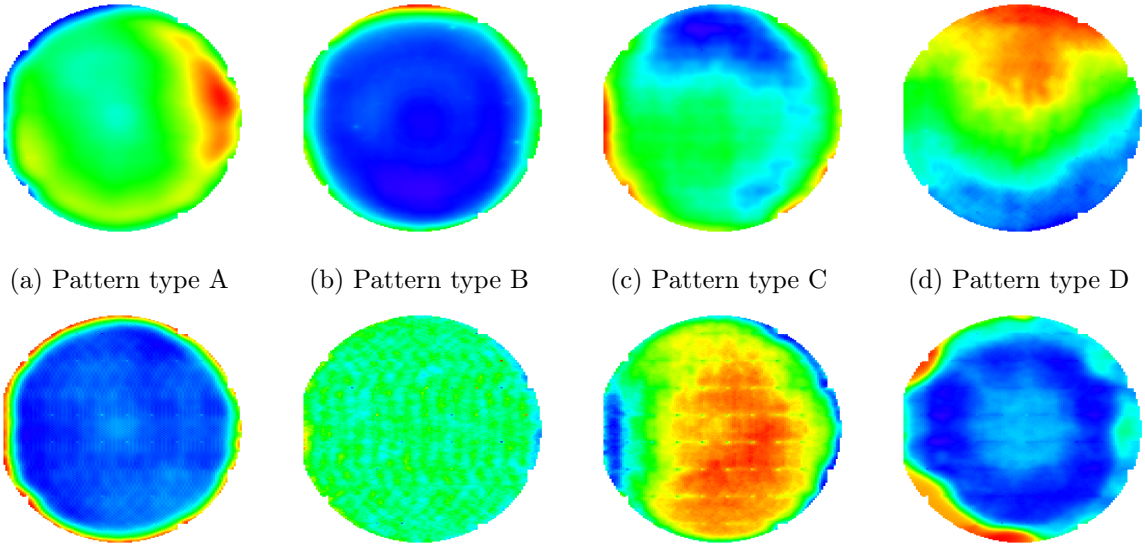
return simulated wafermap

The simulation procedure by Pleschberger [77] can be described by Alg. 2. The procedure accounts for distinct types of distortions, which might occur in practical applications, e.g. outliers, NAs or measurement noise. For details, we refer the reader to the work of Pleschberger.

The suggested procedure was implemented in the statistical programming language R [132]. However, to permit comparable and reproducible results, one dataset generated from the described approach was published as a free dataset by Pleschberger et al. [133]. In the course of this work, the simulated dataset will be denoted as dataset 1.

Each wafermap depicts exactly one pattern. The online dataset comprises a total number of 5000 wafermaps, each of the 5 pattern types is represented on 1000 instances. A typical train-test-split is provided by a subset of 4000 wafermaps (800 per pattern type) for training and 1000 wafermaps (200 per pattern type) for testing. Both datasets are balanced w.r.t. the pattern types. Each wafermap comprises approx. 17500 elements (devices). The simulation parameters (number of outliers, noise ratio, etc.) were sampled randomly for each wafermap. However, the intensity information (mainly covered by the window size of the Gaussian blur) was preserved from the randomized simulation parameters in order to provide a ground truth for the pattern intensity ranking in Section 4.4.2.

The simulated dataset can be used for evaluating feature extraction, pattern type classification and pattern intensity, as well as for the overall wafermap-Health Factor. However, wafer- or



(a) Pattern type A (b) Pattern type B (c) Pattern type C (d) Pattern type D

(e) Pattern type E (f) Pattern type F (g) Pattern type G (h) Pattern type H

Fig. 4.2: Pattern types covered in dataset 2, originating from a real-world semiconductor product.

lot-Health Factors cannot be obtained, since no assignment of multiple wafermaps to one wafer is made in the simulation.

4.2.2 Real-World Datasets

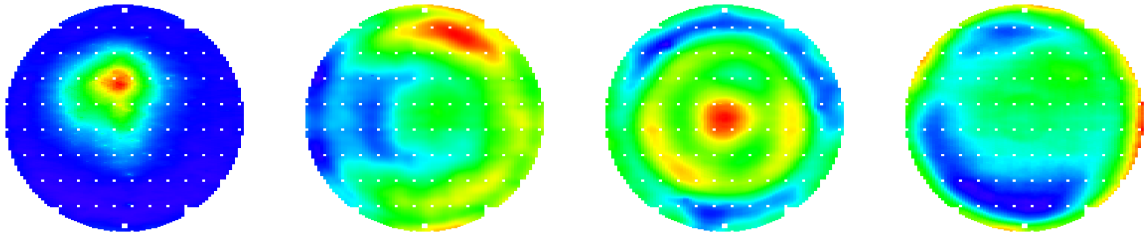
Although large effort was made to construct a synthetic wafer test dataset, the variety of real-world patterns cannot be reached. Hence, it is necessary to evaluate the methods on such datasets, in addition.

In particular, two datasets, dataset 2 and dataset 3 will be exploited for evaluation. Both depict different semiconductor products and hence, consist of a distinct number of devices per wafer. Also the electrical parameters measured and depicted on the wafermaps are different. Nevertheless, the products are similar to each other from the perspective of the manufacturing process chain.

Dataset 2 comprises real-world wafermaps from the same semiconductor product, which was used to create the template used in Section 4.2.1. It consists of 8 pattern types, denoted by pattern type A to H, which are depicted in Fig. 4.2. As in the simulated dataset, all pattern types have specific characteristics, e.g. rotation or resizing of the ROIs. The labeling of the pattern types with this dataset was performed by product experts by taking optical judgment as well as relations between the electrical parameters into account.

In total, each production lot of dataset 2 comprises around 50 wafers, while 21 electrical parameters are measured. Hence, 1050 wafermaps are contained in the wafer test data of each lot. Correlations can be observed between the wafermaps, which originate from the same electrical parameter.

Similarly, dataset 3 contains wafermaps from another semiconductor product, which are optically divided into 4 groups $\mathfrak{A}$ - $\mathfrak{D}$, again labeled by a product expert and depicted in Fig. 4.3. In total, approx. 14000 devices are present on each wafermap within this dataset, i.e. the image



(a) Pattern type $\mathfrak{A}$ (b) Pattern type $\mathfrak{B}$ (c) Pattern type $\mathfrak{C}$ (d) Pattern type $\mathfrak{D}$

Fig. 4.3: Pattern types covered in dataset 3, originating from a real-world semiconductor product.

	A	B	C	D	E	F	G	H	total
training set	100	100	100	100	100	100	100	100	800
test set	147	147	147	147	147	147	147	147	1176
total sample size	568	566	918	568	284	852	284	284	4324

Tab. 4.1: Sample sizes (number of wafermaps) of class A-H in datasets 2, originating from a real-world semiconductor product.

resolution is slightly lower than for the product in dataset 1 and 2. In total, the dataset comprises 6 production lots of 25 wafers, each. The test sequence of this product consists of 25 parameters.

In general, real-world data are not balanced, hence Tab. 4.1 and 4.2 provides the exact sample sizes for each pattern type category in dataset 2 and 3. Furthermore, the train-test-split per class is depicted in the tables. In each run of the experiments, a training and test set of the according pattern type are sampled randomly - multiple runs are performed in order to eliminate influences of the train/test split.

Real-world data yield one major drawback compared to simulations: labels, which are essential to evaluate results of machine learning methods, need to be provided by experts. The generation of manual labelings is associated with large effort for human experts, as well as with subjective decisions and errors. Hence, bad evaluation results might result from both, a bad quality of the method or an unreliably labeled test dataset. This circumstance lowers the significance and expressiveness of the results obtained from evaluations on real-world datasets. However, real-world data are crucial to demonstrate that a system is mature for being deployed in manufacturing.

	$\mathfrak{A}$	$\mathfrak{B}$	$\mathfrak{C}$	$\mathfrak{D}$	total
training set	40	40	40	40	160
test set	45	45	45	45	180
total sample size	194	194	194	97	679

Tab. 4.2: Sample sizes (number of wafermaps) of class $\mathfrak{A}$-$\mathfrak{D}$ in dataset 3, originating from a real-world semiconductor product.

		ground truth	
		1	0
prediction	1	TP	FP
	0	FN	TN

Tab. 4.3: Confusion matrix (contingency table) evaluating the results of a binary classifier.

In order to draw significant conclusions on the quality of the methods, it is essential to take results from both, real-world data and simulated data into account. Therefore, benefits and limitations from both types of data must be considered.

4.3 Evaluation of Pattern Type Classification

Before assessing the concept of the Health Factor as a whole, an evaluation of the single components will be performed. Pattern type classification is investigated particularly in detail for three reasons: firstly, this component has the highest influence on the result of the Health Factor and therefore, will be crucial to obtain accurate results. Secondly, most of the research effort accompanying this thesis (as well as a large part of the scientific contribution) are comprised in the pattern type classifier. As a third aspect, the pattern type classifier consists of distinct elementary steps, hence the evaluation effort is higher than for the other components. Hence, the experiment aims to judge the quality of the pattern type classifier and extends the results presented in Schrunner et al. [18, 19].

4.3.1 Evaluation Measures

From the perspective of machine learning, the quality of the pattern type classifier can be assessed by different metrics. For example, Zhang and Zhou [134] presented a review on multi-label classifiers, where they deployed and explained the following evaluation measures:

- accuracy,
- precision / recall,
- F_β score,
- Receiver operating characteristic (ROC) curve.

In order to define and analyze these evaluation measures, we consider a data sample $\{x_1, \ldots, x_n\}$ with predicted labels $c(x_i) \in C = \{c_1, \ldots, c_N\}$ and ground truth labels $c^*(x_i) \in C$ for all $i = 1, \ldots, n$. Hence, the four major key values *true positive (TP)*, *false positive (FP)*, *true negative (TN)* and *false negative (FN)* can be obtained as depicted in Tab. 4.3 for a binary classification problem with classes 1 and 0.

		ground truth			
		$\dots$	$i-1$	i	$i+1$ $\dots$
prediction	$\dots$			FN_i	
	$i-1$				
	i		FP_i	TP_i	FP_i
	$i+1$			FN_i	
	$\dots$				

Tab. 4.4: Confusion matrix evaluating the results of a multiclass classifier. While all row elements (except for the diagonal element) contribute to FP_i, all column elements (except for the diagonal element) contribute to FN_i.

Since more than 2 classes might be of interest, this concept can be extended to the multiclass case, where a true positive TP_i , false positive FP_i and false negative FN_i value can be defined for each class $i \in \{1, \dots, N\}$. These are defined as follows:

$$TP_i = |\{x_j : c(x_j) = c^*(x_j) = i\}| \tag{4.1}$$

$$FN_i = |\{x_j : c(x_j) \neq i, c^*(x_j) = i\}| \tag{4.2}$$

$$FP_i = |\{x_j : c(x_j) = i, c^*(x_j) \neq i\}| \tag{4.3}$$

Hence, it is preferred that the main diagonal of the confusion matrix contains most of the classified samples. Graphically, the according confusion matrix is illustrated in Tab. 4.4. Note that a TN class does not exist in the multiclass case, since each class is equally assumed as positive.

Accuracy The most intuitive evaluation measure is the accuracy, i.e. the ratio of correctly labeled instances in the test set. Hence, the accuracy is defined as follows:

$$Acc(TP, FP, FN, TN) = \frac{TP + TN}{TP + FP + FN + TN}, \tag{4.4}$$

or, in the multiclass case,

$$Acc(TP_1, \dots, FP_1, \dots, FN_1, \dots) = \sum_{i=1}^{N} \frac{TP_i}{n} = \frac{\sum_{i=1}^{N} TP_i}{n}. \tag{4.5}$$

The accuracy is bounded and ranges between 0 and 1; 0 denotes a completely wrong assignment, whereas 1 is obtained from a perfect classifier.

However, it is obvious from the following example that accuracy lacks significance in case of unbalanced classes: assume that the test set of class A contains 1000 samples, while the test set of class B contains only 10. In case that a classifier assigns each new instance to class A (which is obviously incorrect), there will be 1000 samples labeled correctly as class A (TP) and

10 samples labeled as false positives (FP). FN and TN are empty in this situation. Hence, the accuracy Acc is obtained as

$$Acc(TP, TN) = \frac{1000}{1010} = 0.99,$$

which suggests a perfect prediction, although the results of the classifier are inadequate.

Precision, recall and F_β score Since accuracy takes rather the correctly labeled data than the type of error into account, precision and recall are investigated. Precision judges the type I error (error by labeling too many instances as positives) and recall accounts for the type II error (error by labeling too many instances as negatives). Hence, a bad quality of the result can be traced back to one of these two aspects.

In detail, the precision Pre and the recall Rec are defined in the following way in a two-class setting:

$$Pre(TP, FP, FN, TN) = \frac{TP}{TP + FP} \tag{4.6}$$

$$Rec(TP, FP, FN, TN) = \frac{TP}{TP + FN}. \tag{4.7}$$

However, practical applications demand to keep both types of errors in an acceptable range. Thus, it is necessary to combine these into a single parameter, which performs a trade-off between both metrics. In contrast to the accuracy, this measure is also able to deal with unbalanced test sets: the F_β score. Given a parameter $\beta > 0$ the F_β score is defined as follows for the binary classification problem:

$$F_\beta(TP, FP, FN, TN) = \left(1 + \beta^2\right) \frac{Pre(TP, FP, FN, TN) \cdot Rec(TP, FP, FN, TN)}{(\beta^2 Pre(TP, FP, FN, TN)) + Rec(TP, FP, FN, TN)}, \tag{4.8}$$

whereas the most prominent representative of this group is the F_1 score

$$F_1(TP, FP, FN, TN) = 2\frac{Pre(TP, FP, FN, TN) \cdot Rec(TP, FP, FN, TN)}{Pre(TP, FP, FN, TN) + Rec(TP, FP, FN, TN)}. \tag{4.9}$$

The F_1 score is the harmonic mean between precision and recall, hence represents the balanced version of the F-score. Similar to the accuracy, the F_1 score is bounded between 0 and 1, where 1 denotes perfect results.

Multiple options for extending the F_1 score to a multiclass setting exist. The most prominent include micro-averaging or macro-averaging, see e.g. Zhang and Zhou [118], while micro-averaging calculates an average between the classes within the calculation of F_1 value, macro averaging F_1 scores result from averaging the class-wise scores by calculating the F_1 score for each class i separately. Formally, this means that

$$F_1^{micro}(TP_1, \dots, FP_1, \dots, FN_1, \dots) = F_1\left(\sum_{i=1}^{N} TP_i, \sum_{i=1}^{N} FP_i, \sum_{i=1}^{N} FN_i\right), \tag{4.10}$$

$$F_1^{macro}(TP_1, \dots, FP_1, \dots, FN_1, \dots) = \sum_{i=1}^{N} F_1(TP_i, FP_i, FN_i). \tag{4.11}$$

setup no.	preprocessing	feature extraction			machine learning
	MRF	LBP	RLBP	HOG	classifier
1.a	**yes**	**yes**	**yes**	**yes**	**Naive Bayes**
1.b	no	yes	yes	yes	Naive Bayes
1.c	yes	yes	no	no	Naive Bayes
1.d	yes	yes	yes	no	Naive Bayes
1.e	yes	no	no	yes	Naive Bayes
1.f	yes	yes	yes	yes	SVM (linear kernel)
1.g	yes	yes	yes	yes	SVM (polynomial kernel)
1.h	yes	yes	yes	yes	decision tree
1.i	yes	yes	yes	yes	random forest
1.j	yes	yes	yes	yes	logistic regression (OVO)
1.k	yes	yes	yes	yes	SSC-UC

Tab. 4.5: Experimental setups for experiment 1. 1.a (bold) represents the default setup, which will be used for comparison.

4.3.2 Experimental Setup

The major steps of the investigated pattern type classification procedure are

- data preparation and preprocessing (see Section 3.1),
- feature extraction (see Section 3.2) and
- supervised or semi-supervised machine learning (see Section 3.3.2).

Unfortunately, it is hardly possible to evaluate these steps separately while eliminating all influences from the other methods. Single results for MRF are presented in Schrunner et al. [13], those for LBP and RLBP are presented in Santos et al. [15]. However, both of these works apply additional methods (e.g. distance measures, clustering, etc.), which influence the outcomes. The evaluation performed in this work will be done in a combined setting, observing all of these 3 aspects.

The experiment is conducted on all 3 datasets (simulated and real-world data) depicted in Section 4.2. We define a standard setting, which will serve as a reference. The Naive Bayes classifier is chosen in this standard setting for two reasons: firstly, this classifier is generative, which is beneficial for interpreting the results and understanding the influences of the decision. Secondly, the semi-supervised adaption of the classifier to integrate unknown classes is based on this setup. The logistic regression and the SVM are used with the one-vs-one (OVO) setting, while all other approaches are applied in their multiclass version. In summary, the experimental setups used in this evaluation are depicted in Tab. 4.5. 1.a – 1.j and refer to supervised methods in order to perform pattern type classification, while 1.k contains the Semi-Supervised Classifier with Unknown Classes (SSC-UC) approach.

Upon varying the setup for preprocessing, feature extraction and supervised machine learning, the SSC-UC presented in Section 3.3.3 will be separated from the other setups during evaluations:

setup no.	pattern type 1	2	3	4	5
1.a	**1.000**	0.994	**1.000**	**1.000**	0.994
1.b	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
1.c	0.925	0.753	**1.000**	0.843	0.727
1.d	0.977	0.996	**1.000**	0.971	**1.000**
1.e	**1.000**	0.997	0.979	0.982	**1.000**
1.f	**1.000**	0.964	**1.000**	0.981	0.984
1.g	0.867	0.775	0.978	0.794	0.959
1.h	0.991	0.996	**1.000**	0.991	0.992
1.i	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
1.j	**1.000**	0.999	**1.000**	0.999	**1.000**

Tab. 4.6: Experimental results for experiment 1 on dataset 1. The F_1 scores are plotted for each experimental setup from Tab. 4.5 and for each pattern type. The highest scores achieved in each column are marked bold.

since SSC-UC is tailored to distinguish multiple known and unknown classes, the setup will be changed, such that for dataset 1 and experimental setup 1.k, only a subset of the pattern types is present in the labeled training data, while all 5 pattern types appear in the unlabeled training and test data. Hence, the evaluation will be made using a macro-averaging F_1 score over the known classes, while the unknown classes are aggregated to a new class *unknown* (0). In addition, accuracy values will be provided to get a closer insight into the results.

In particular, the number of classes known via labeled samples at training time is denoted by k for experiment 1.k. The values for k are varied by $\{1, 2, 3, 4\}$ for the simulated dataset. The special case with $k = 5$ would result in a setup without unknown classes, which can be tackled by the other methods evaluated in experiments 1.f – 1.j, while $k = 0$ would result in an unsupervised problem. For the real-world datasets, the SSC-UC method is evaluated on a subset of $k = 6$ classes (dataset 2) and $k = 3$ classes (dataset 3). In each of the 100 performed runs, the k known classes are sampled randomly from the full set of classes of each dataset.

4.3.3 Results

The experimental results for experiment 1, depicting the F_1 score for each pattern type of each dataset, are provided in Tab. 4.6 – 4.8. Furthermore, Tab. 4.9 contains the macro-averaged F_1 scores for each dataset in the different experimental setups.

Supervised settings In general, almost all experimental setups achieve results with an acceptable F_1 score - however, differences can be clearly observed although most scores range between 0.85 and 1.0.

Firstly, the MRF approach performs slightly worse on dataset 1 and 3, but massively improves the results in dataset 2 (especially pattern types B, C and D). Hence, we conclude that there might be some pattern types, which rely on smoothing to perform recognition, while for others, smoothing has a slightly negative influence. Since the negative influence appears to be in a low

setup no.	pattern type A	B	C	D	E	F	G	H
1.a	0.998	0.994	0.868	0.981	**1.000**	0.975	0.979	0.807
1.b	0.998	0.777	0.728	0.766	0.926	0.972	0.982	0.946
1.c	0.917	0.585	0.530	0.671	0.880	0.992	0.921	0.762
1.d	0.963	0.597	0.532	0.712	0.914	0.993	0.984	0.842
1.e	0.576	0.662	0.657	0.449	0.734	0.875	0.551	0.750
1.f	**1.000**	0.965	0.943	0.969	0.997	0.979	0.959	0.910
1.g	0.933	0.728	0.403	0.788	0.990	0.956	0.807	0.675
1.h	0.956	0.688	0.780	0.758	0.951	0.950	0.867	0.804
1.i	**1.000**	0.994	0.997	0.990	**1.000**	0.995	**0.993**	0.990
1.j	0.999	**0.997**	**0.998**	**0.994**	**1.000**	**0.996**	**0.993**	**0.997**

Tab. 4.7: Experimental results for experiment 1 on dataset 2. The F_1 scores are plotted for each experimental setup from Tab. 4.5 and for each pattern type. The highest scores achieved in each column are marked bold.

setup no.	pattern type $\mathfrak{A}$	$\mathfrak{B}$	$\mathfrak{C}$	$\mathfrak{D}$
1.a	0.998	0.935	0.922	0.976
1.b	**1.000**	**0.998**	**0.985**	0.986
1.c	**1.000**	0.848	0.791	**0.994**
1.d	0.997	0.928	0.902	0.947
1.e	0.953	0.883	0.900	0.953
1.f	**1.000**	0.870	0.873	0.954
1.g	0.991	0.709	0.783	0.675
1.h	0.980	0.837	0.819	0.853
1.i	**1.000**	0.970	0.965	0.986
1.j	0.932	0.879	0.864	0.900

Tab. 4.8: Experimental results for experiment 1 on dataset 3. The F_1 scores are plotted for each experimental setup from Tab. 4.5 and for each pattern type. The highest scores achieved in each column are marked bold.

setup no.	dataset 1	2	3
1.a	0.997	0.950	0.958
1.b	**1.000**	0.887	**0.992**
1.c	0.850	0.782	0.896
1.d	0.989	0.817	0.943
1.e	0.992	0.657	0.922
1.f	0.986	0.965	0.924
1.g	0.874	0.785	0.851
1.h	0.994	0.844	0.872
1.i	**1.000**	0.995	0.980
1.j	**1.000**	**0.997**	0.894

Tab. 4.9: Macro-averaged F_1 scores for the results of experiment 1 in all 3 datasets.

range compared to the benefit regarding these 3 pattern types, we recommend to integrate the MRF approach into the default preprocessing pipeline.

Secondly, we observe that feature extraction has a high impact on the result: none of the subsets provided in experimental setups 1.c - 1.e can constantly beat this default setup, where LBP, RLBP and HOG features are used. While for the patterns in dataset 1, it seems that e.g. HOG carries enough information to distinguish the 5 patterns, it is obvious that patterns A-H in dataset 2 cannot be distinguished correctly by any of the lower-dimensional combinations.

As a third aspect, we observe that the choice of the multiclass classifier mostly dominates the F_1 score within the range of 0.9 to 1.0. While in general, the polynomial SVM version as well as the decision tree perform worse than the other methods in the comparison, the random forest and the logistic regression perform slightly better than the standard setting using a Naive Bayes classifier. Concerning the quality of the results, the random forest dominates dataset 1, while logistic regression performs best on dataset 2. Thus, we suggest that a higher total number of classes (see dataset 2) can be covered by the pairwise OVO comparisons in a better way than by a random forest. However, in dataset 3, both the random forest and the Naive Bayes perform very well.

Semi-supervised setting Apart from the purely supervised setups, the SSC-UC method is evaluated in Tab. 4.10. Here, the evaluation on dataset 1 is performed for distinct settings of the parameter k, denoting the number of known classes. For the real-world datasets, k is set to approx. 75 % of the total number of classes, i.e. $k = 6$ for dataset 2 and $k = 3$ for dataset 3. For each of the respective datasets, the unlabeled training set as well as the test set contain elements from all available classes (known and unknown).

In general, the quality of the SSC-UC results is good on the simulated dataset, which is in accordance with the behavior of the related supervised classifiers evaluated in the settings 1.a – 1.j. In particular, the F_1 scores and accuracies obtained from the classifier are similar for different settings of k, i.e. for a distinct portion of known and unknown classes. Note that on the

	dataset 1				dataset 2	dataset 3
k	1	2	3	4	6	3
Acc	0.999	0.998	0.996	0.999	0.771	0.890
F_1	0.993	0.992	0.989	0.998	0.684	0.838

Tab. 4.10: Results for experimental setup 1.k, i.e. evaluating the SSC-UC method with two unknown classes in each dataset. Acc and F_1 are evaluated upon all known classes (macro-averaging accuracy and F_1 score). k denotes the number of known classes during training.

one hand, the problem setup becomes more challenging for an increasing number of unknown classes, since the inhomogeneity of the residual class is larger. On the other hand, a higher number of known classes results in a larger choice regarding the prediction labels, which is also a factor contributing to the difficulty of the problem. In addition, the human classification of the training data (ground truth labels) might be erroneous. Hence, a clear ranking of the difficulty of the problem setups depending on k is not possible.

However, for the real-world datasets, the evaluation measures depict a worse behavior. This might be caused by different reasons: firstly, the real-world dataset 2 contains a higher total number of pattern types, i.e. a discrimination is (in general) more difficult. Secondly, real-world wafermaps show background patterns and additional influences, which are not present in the simulation. Hence, it is more difficult for the algorithm to discriminate between pattern-related variations and spurious correlations - in particular, this problem arises when no labeled training data is available, which is the case for the unknown classes in the experiment, similar to the problems arising with unsupervised learning.

Although showing lower values than in the supervised settings, where all labeled data were known, SSC-UC solved the problem of detecting elements from previously unseen classes and, at the same time, performs multiclass classification for the known elements. As shown by experiments on UCI datasets in Schrunner et al. [17], the method outperforms state-of-the-art methods in many cases.

4.4 Evaluation of Pattern Intensity Quantification

Experiment 2 focusses on the evaluation of the pattern intensity quantifier and is based on Jenul [128]. Although having a minor influence on the Health Factor compared to the pattern type classifier, intensity quantification directly influences the result of the Health Factor. Intensity quantification is, by definition an unsupervised problem - hence, an experimental setup for unsupervised learning is applied.

4.4.1 Evaluation Measures

The intensity quantifier is a function that delivers values on a continuous $[0, 1]$ scale for each wafermap. Furthermore, the absolute value is less informative than the relative ranking of the intensity among wafermaps showing the same pattern type. As a result, the measures deployed to assess pattern intensity are key values, which compare a ranking based on the ground truth values to a ranking obtained from the intensity quantifier.

In statistics, two prominent evaluation measures perform this task:

- Spearman's rank correlation coefficient (Spearman's ρ_S) and
- Kendall's τ.

Spearman's ρ_S Given two equally-sized sets of realizations $\{x_1, \dots, x_n\}$ and $\{y_1, \dots, y_n\}$ of random variables X and Y, we denote the rank of each element x_i by $r(x_i)$ (the same is assumed for y_j). Spearman's rank correlation coefficient ρ_S is defined as the Pearson correlation cofficient of $\{r(x_1), \dots, r(x_n)\}$ and $\{r(y_1), \dots, r(y_n)\}$, i.e.

$$\rho_S = \frac{\mathrm{Cov}(\{r(x_1), \dots, r(x_n)\}, \{r(y_1), \dots, r(y_n)\})}{\sqrt{\mathrm{Var}(\{r(x_1), \dots, r(x_n)\}) \cdot \mathrm{Var}(\{r(y_1), \dots, r(y_n)\})}}. \tag{4.12}$$

In analogy to the Pearson correlation coefficient ρ, Spearman's ρ_S delivers values between -1 and 1, where 1 denotes that $x_1, \dots, x_n$ and $y_1, \dots, y_n$ follow the same order, while -1 indicates that the sets are ordered in a reverse way. A rank correlation around 0 is obtained, if the rankings of $x_1, \dots, x_n$ and $y_1, \dots, y_n$ are uncorrelated.

Since Spearman's rank correlation coefficient merely takes the ranks of the data into account, it is invariant w.r.t. any strictly monotonic transformation of the data scale. Furthermore, the measure is mainly intended for ordinal data with no tied pairs.

Kendall's τ The idea of Kendall's τ is similar to Spearman's ρ_S, whereas Kendall's τ takes an interval scale of the dataset into account. As before, we consider two sets $\{x_1, \dots, x_n\}$ and $\{y_1, \dots, y_n\}$ and investigate their ranks $r(x_i)$ and $r(y_j)$, respectively. The number of concordant and discordant pairs between the two sets are calculated, defined as follows for $i < j$:

$$(x_i, y_i) \text{ and } (x_j, y_j) \text{ concordant} \quad :\Leftrightarrow \quad \begin{cases} x_i > x_j \text{ and } y_i > y_j, \text{ or} \\ x_i < x_j \text{ and } y_i < y_j, \end{cases} \tag{4.13}$$

$$(x_i, y_i) \text{ and } (x_j, y_j) \text{ discordant} \quad :\Leftrightarrow \quad \begin{cases} x_i > x_j \text{ and } y_i < y_j, \text{ or} \\ x_i < x_j \text{ and } y_i > y_j, \end{cases} \tag{4.14}$$

Pairs with $x_i = x_j$ or $y_i = y_j$ are considered to show neither of the two characteristics.

In case that the ranks of both sets match perfectly, the number of concordant pairs will be high, while no discordant pair exists. In the opposite situation, where no concordant pair exists, the sets are ordered in a reverse way. Accordingly, the formula for Kendall's τ compares the difference between concordant and discordant pairs to the total number of pairs in the dataset given $i < j$, i.e. $\frac{n(n-1)}{2}$:

$$\tau = \frac{|\{(i,j) : (x_i, y_i) \text{ and } (x_j, y_j) \text{ concordant}\}| - |\{(i,j) : (x_i, y_i) \text{ and } (x_j, y_j) \text{ discordant}\}|}{\frac{n(n-1)}{2}}. \tag{4.15}$$

Again, the value ranges between -1 and 1 and can be interpreted in the same way as Spearman's ρ_S. Kendall's τ is preferred e.g. for smaller datasets or in case that multiple data points with

setup no.	features / stand-alone method	combination
2.a	**all**	**feature selection**
2.b	all	PCA
2.c	Moran's I	none
2.d	Spatial entropy	none

Tab. 4.11: Experimental setups for experiment 2. 2.a (bold) represents the default setup, which will be used for comparison.

the same score exist. However, the computational complexity is worse than for Spearman's ρ_S. For the purpose of evaluation, both metrics will be evaluated in the following.

4.4.2 Experimental Setup

In order to assess the quality of the intensity quantifier, a set of labeled (here: ranked) wafermaps is required. Attempts to perform an excessive labeling of the real-world datasets by experts failed during our investigations, since an optical ranking is time-intensive and prone to errors and subjective decisions in larger datasets (probands assigned different orders to wafermaps of similar intensity values).

Hence, the evaluation of the intensity concept will be performed on the simulated dataset (dataset 1), where the ground truth intensity can be derived from the randomized simulation parameters and one small subset of dataset 2. Depending on the pattern type of the simulation, especially the kernel size used for Gaussian blur strongly influences the underlying intensity.

However, a conceptional problem exists when analyzing the gradient pattern (pattern type 3): the first aspect of intensity is x-y-intensity, i.e. the spatial distribution of the pattern on the wafer. By definition, the gradient pattern spreads over the whole wafermap, thus no graduation is possible in this respect. Furthermore, the second aspect is the z-intensity, i.e. the higher or lower ascent or descent of the pattern. Since the gradient pattern is characterized by linearly increasing values along any major direction of the wafermap, the image gradient will be nearly constant over the wafermap. As the gradient ∇ is a linear functional, i.e. $\alpha \cdot \nabla f = \nabla(\alpha \cdot f)$ for any differentiable, real-valued function f and any scalar $\alpha \in \mathbb{R}$, it is clear that a normalization of the original dataset will remove any information from the z-intensity. Hence, the intensity of this pattern cannot be quantified. Since the patterns do not cover the whole wafermap for all other pattern types, intensity differences are present due to distinct x-y-expansions of the patterns.

In conclusion, the following evaluation will be done for pattern types 1, 2, 4 and 5 for the simulated dataset (dataset 1). Furthermore, the real-world subset utilized for evaluation contains wafermaps of pattern type H in dataset 2. This pattern type is considered the main pattern of interest in the respective dataset and was successfully labeled by an expert.

For all datasets, the rank correlation values (Spearman's ρ_S and Kendall's τ) will be compared between the predicted labeling obtained from the intensity quantification procedure and the ground truth (or expert-based) labeling.

The parameter settings, which are used for the intensity evaluation are explained in Tab. 4.11. Experimental setups 2.a and 2.b represent the combination approaches, based on a feature ma-

	dataset 1				dataset 2
	1	2	4	5	H
2.a	**0.986**	0.540	**0.996**	**0.995**	**0.817**
2.b	0.958	0.471	**0.996**	**0.995**	0.306
2.c	0.981	**0.942**	**0.996**	**0.995**	0.190
2.d	0.938	0.443	**0.996**	**0.995**	-0.418

(a) Spearman's ρ_S

	dataset 1				dataset 2
	1	2	4	5	H
2.a	**0.907**	0.453	0.961	**0.955**	**0.688**
2.b	0.820	0.386	0.957	0.954	0.254
2.c	0.894	**0.782**	0.961	0.955	0.158
2.d	0.775	0.335	**0.962**	0.953	-0.341

(b) Kendall's τ

Tab. 4.12: Experimental results for experiment 2, grouped by pattern types. In the table, Spearman's ρ_S (4.12a) and Kendall's τ (4.12b) are depicted for each experimental setup from Tab. 4.11.

trix, which covers Moran's I, Spatial entropy (both in global and local versions on ROIs), as well as the size of the ROI, the normalized difference between the ROI median and the global median. Further, 2.c and 2.d depict the performance of the global Moran's I and Spatial entropy as stand-alone methods. Evaluations using a refined set of simulated data are provided in [128].

4.4.3 Results

An overview of the performance comparison of each method is provided in Tab. 4.12. The results demonstrate that the simulated pattern types can be described in an accurate way by the intensity measures, while the real-world dataset achieves worse results regarding Spearman's ρ_S and Kendall's τ. Two reasons might influence this behavior: firstly, real-world patterns are affected by hardly visible background structures, which distort the statistical autocorrelation measures and lead to wrong orders. Although image segmentation can reduce the effect of these structures, it is not able to completely prevent the evaluation from such influences. Secondly, the ground truth for the real-world datasets is based on human assessment, while the ground truth information for the simulated datasets is extracted from the simulation parameters - as a result, erroneous or subjective rankings of the ground truth might provoke a bad rank correlation, although the intensity values represent the "correct" order. Furthermore, the fact that Gaussian white noise is added to the simulations, while measurement noise of real-world data might violate this distribution assumption, has an impact on the results.

With regard to the comparison of the methods deployed to quantify the intensity of wafermaps, it can be observed that the stand-alone version of Moran's I outperforms Spatial entropy for most pattern types, in particular for pattern type 2 (spot pattern) in the simulated dataset.

However, for the real-world dataset, Spatial entropy has a higher absolute correlation, but with a negative sign. According to Jenul [128], this is caused by the fact that Spatial entropy is positively correlated with an increasing z-intensity, but negatively correlated with an increasing x-y-intensity - hence, the type of intensity dominating the pattern type is responsible for the positive or negative sign of the rank correlation. A similar issue can occur, if PCA is used (although it was not observed in the given experiments): since PCA is invariant w.r.t. the sign of the eigenvectors, it is possible that an inverse intensity order is obtained. Finally, the overall quality of the default setup, i.e. the feature selection is better than any other setup. Hence, this version will be used for evaluating the Health Factor in the following.

With regard to the evaluation metrics applied to assess the results, one can observe that all methods achieve the same value of Spearman's ρ_S for pattern types 4 and 5 of the simulated dataset, while the values of Kendall's τ differ. While the quality of the results is similar among all methods, Kendall's τ is more sensitive to minor differences, which cannot be covered by Spearman's ρ_S within 3 decimal places.

4.5 Health Factor Evaluation

On top of evaluating the effect of each single component of the Health Factor system, a validation of the full concept is necessary. The validation concept is similar to the experimental part of our previous work [18, 19]. All three datasets will be evaluated for this purpose. However, validating the obtained values via a ground truth is hardly possible, since the concept is novel and contains additional information, which was not covered by preceding key values. Hence, the validation of the Health Factor will be done as a case study, where we evaluate whether those wafermaps, which are categorized as specifically critical, can be covered by the Health Factor value or not.

4.5.1 Experimental Setup and Evaluation

In order to evaluate the Health Factor as a whole, all three components need to be provided. The pattern type classifier and the intensity quantifier will be used in their default versions as provided in the preceding evaluation of experiments 1 and 2. In order to calculate the Health Factor, the version of hard classification will be deployed, since this version conforms with the consistency property discussed in Section 3.4.5.

Since the indicator delivers values on a continuous scale, but an expert will rather be interested in a binary decision (e.g. is an alert triggered or not), a threshold t is applied on the Health Factor. Using this threshold, the resulting values are mapped to a 0-1 scale. Thereby, 1 denotes that the Health Factor value is greater than t, and 0, else. Given a test dataset containing wafermaps with ground truth classifications as 0 (uncritical) and 1 (critical), a binary classification problem is obtained. Hence, ordinary evaluation metrics such as the F_1 score can be applied for evaluation.

In the evaluations, the default settings for the pattern type classifier, as well as for the intensity, presented in Tab. 4.5 and Tab. 4.11 are applied. Furthermore, the criticality values specified by the expert for each dataset are depicted in Tab. 4.13. For each intensity value, we vary the Health Factor threshold t in a range between 0 and 1 in steps of size 0.1. In the settings, 3.a and 3.g correspond to the settings presented in Schrunner et al. [18].

In these experiments, note that only one setup is used for each of the real-world datasets, since both of these datasets contain merely one pattern type, which should be tracked. However, the

setup no.	pattern type				
	1	2	3	4	5
3.a	1	0.5	0	0.5	0
3.b	0	0	0	0	1
3.c	0	0	1	0	1
3.d	1	1	0	1	1
3.e	1	1	1	1	1

(a) Criticality values for dataset 1.

setup no.	pattern type							
	A	B	C	D	E	F	G	H
3.f	0	0	0	0	0	0	0	1

(b) Criticality values for dataset 2.

setup no.	pattern type			
	$\mathfrak{A}$	$\mathfrak{B}$	$\mathfrak{C}$	$\mathfrak{D}$
3.g	1	0	0	0

(c) Criticality values for dataset 3.

Tab. 4.13: Experimental setups for experiment 3. Different criticality settings are set in each experimental setup 3.a – 3.g.

behavior of the Health Factor in the presence of multiple critical pattern types is evaluated on the simulated dataset 1. Furthermore, experiment 3.a evaluates the case that criticality values unlike 0 and 1 are specified. Hence, the spectrum of different situations, where the Health Factor is applicable, is demonstrated.

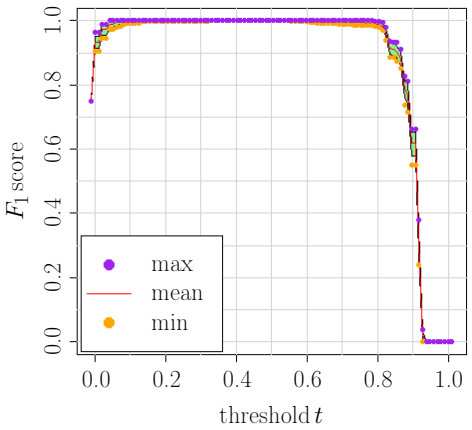
4.5.2 Results

The results for the Health Factor procedure are provided in Fig. 4.4 for the simulated datasets, as well as in Fig. 4.5 for the real-world datasets. In each plot, the achieved F_1 scores are plotted against the threshold value t.

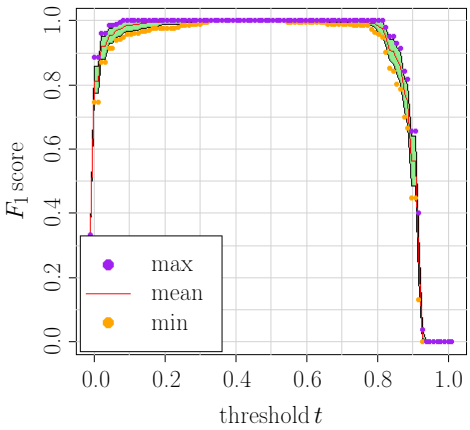
As could be expected from the good results in the pattern type and intensity evaluations from Section 4.3.2 and 4.4.2, the Health Factor achieved accurate results for the simulated dataset. Obviously, if the threshold is set to a very low value, the performance decreases, since almost all instances have Health Factor values above the threshold t and are predicted as critical (1). Hence, the number of false positives (FP) increases and leads to a bad precision. On the other hand, if the threshold t is set to a high value, most instances are predicted as uncritical (0), which leads to a high number of false negatives (FN), i.e. a high recall. Hence, the best F_1 scores are achieved by a trade-off between the two extremes. However, as the plateau of the F_1 function is broad, the Health Factor is rather insensitive to small variations of the threshold parameter t.

In the last criticality setting of the simulated dataset, all pattern types (and hence, all instances) are assumed to be critical, hence, the Health Factor already achieves perfect results, if the threshold is set to 0.

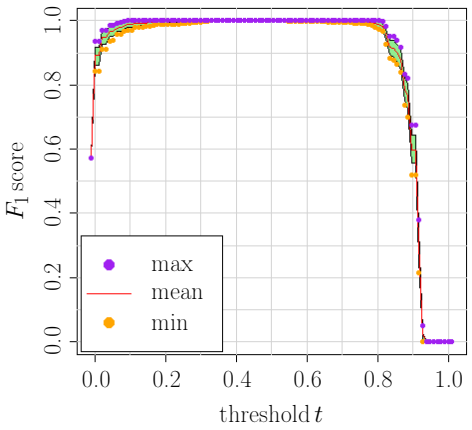
With regard to the real-world datasets, the confidence belt is broader, since more variations result from the uncertainty of the pattern type classifier. Hence, it is possible that misclassifications have a negative effect on the result. Nevertheless, the behavior at an increasing threshold value t



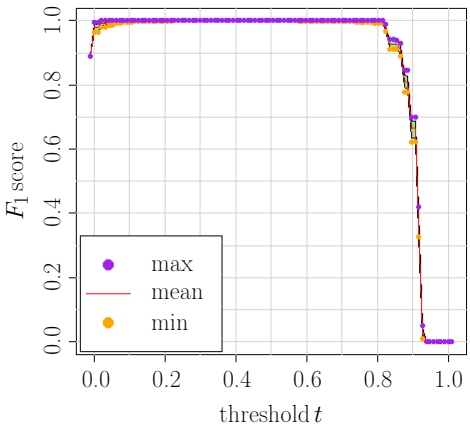
(a) F_1 plot of experiment 3.a.



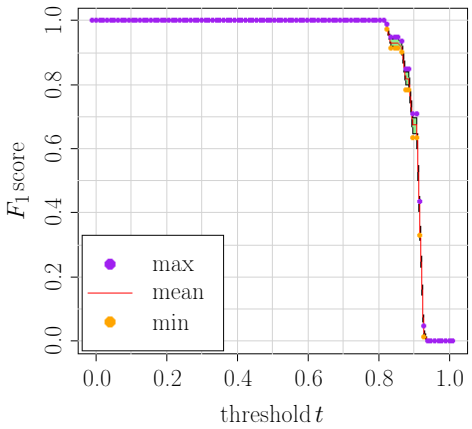
(b) F_1 plot of experiment 3.b.



(c) F_1 plot of experiment 3.c.

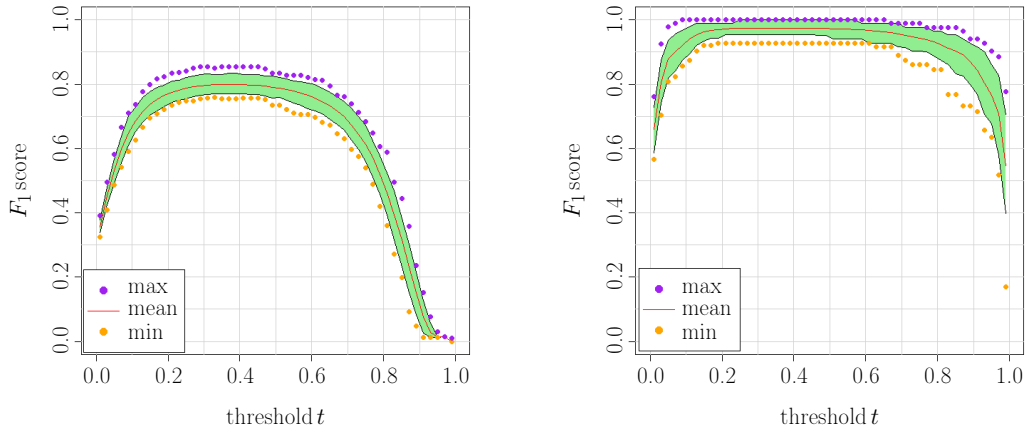


(d) F_1 plot of experiment 3.d.



(e) F_1 plot of experiment 3.e.

Fig. 4.4: Plots of the 90 % confidence belt of the F_1 scores obtained from the Health Factor at a varying threshold t. The results are provided for the experimental setups (a) – (e) on the simulated dataset 1.



(a) F_1 plot of experiment 3.f.

(b) F_1 plot of experiment 3.g.

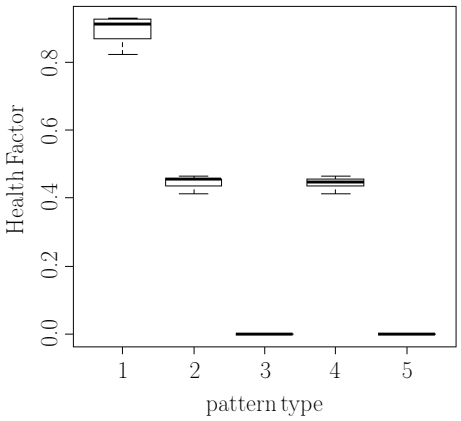
Fig. 4.5: Plots of the 90 % confidence belt of the F_1 scores obtained from the Health Factor at a varying threshold t. The results are provided for the experimental setups (g) – (h) on the real-world datasets 2 and 3.

is similar. For all real-world datasets in this evaluation, only one critical pattern type is present in the dataset.

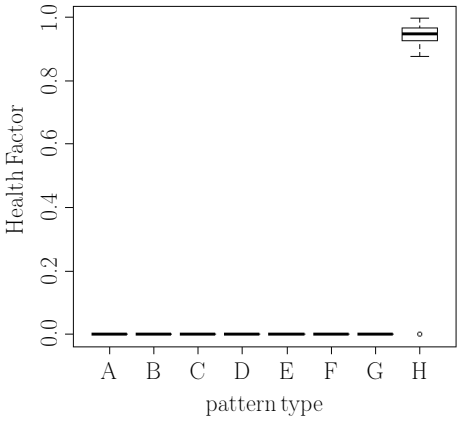
Although accurate F_1 scores can be obtained from the two real-world datasets, it is obvious that the Health Factor performs worse on dataset 2 than on dataset 3. This might be due to the higher number of pattern types available in this dataset or to the similarity of pattern types, leading to misclassifications. This result is in accordance with those presented in the evaluation of the pattern type classifier, where especially pattern types C and H cannot be recognized in all cases. As pattern type H is the critical pattern type of dataset 2, it is obvious that misclassifications are responsible for the lower F_1 score.

In addition to the F_1 scores, Fig. 4.6 provides the boxplots of the original Health Factor values per pattern type for experimental scenarios 3.a, 3.f and 3.g. In the plots, the grouping is performed according to the ground truth pattern types. Obviously, the distributions of the Health Factor values are different between the pattern types, which have distinct criticality levels assigned. The variation of the Health Factor values within the groups are due to variations of the intensity, as well as uncertainties regarding the pattern type. However, most wafermaps depicting uncritical pattern types have Health Factor values around 0, while those with critical process pattern types are significantly higher than 0. Hence, the concept of the Health Factor fulfills its major requirements. Single data points (e.g. in pattern H of 3.f) are misclassified and thus, provoke a Health Factor value of 0. However, these events can be regarded as outliers, since the pattern type classifier proved to be accurate for most pattern types according to experiment 1.

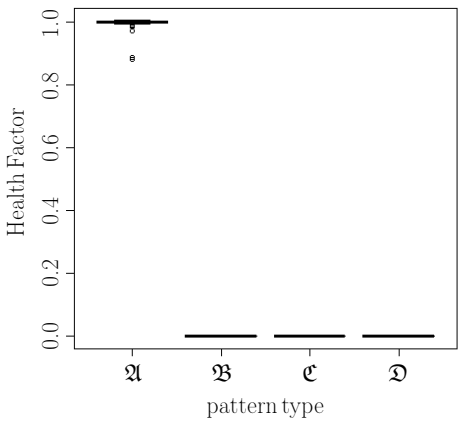
In summary, the results showed numerically that the Health Factor concept is a valid measure to assess the level of concern associated with a wafermap based on the measurements from the electrical wafer test procedure.



(a) Health Factor values per pattern type for experimental setup 3.a.



(b) Health Factor values per pattern type for experimental setup 3.f.



(c) Health Factor values per pattern type for experimental setup 3.g.

Fig. 4.6: Boxplots of the Health Factor values per pattern type for one experimental setup per dataset. The groups on the x-axis represent the ground truth pattern type.

4.6 Discussion

Analyzing the preceding experiments provided insights into the behavior of the suggested Health Factor, as well as its machine learning components. Hence, higher-level aspects, which were not part of the case studies will be discussed, as well as benefits and limitations will be depicted, further. In particular, the validity of the assumptions stated in the introduction, Assumption 1 and 2 (Section 2.3.1), will be analyzed as well as the runtime and the complexity of the methods. Furthermore, interpretability and flexibility of the concept are two important aspects for industry, which will be investigated.

The general concept of the Health Factor is novel to semiconductor industry, including the problem definition for analog data, the decomposition into pattern type, pattern intensity and pattern criticality, as well as the mathematical derivation based on statistical decision theory. Hence, it is worth analyzing benefits and limitations of the concept and those of the components separately.

4.6.1 Benefits and Limitations of the Concept

In general, the concept of the Health Factor is able to fulfill the major requirements stated in the introduction, i.e. it provides a decision support system for experts, based on a sophisticated mathematical framework. For this purpose, it makes use of machine learning and enables to exploit knowledge from data and human experts. Furthermore, the concept demonstrates how machine learning systems can be used in automated manufacturing environments, which goes in line with major demands in industry 4.0.

Exploiting analog data One of the major advantages of the proposed Health Factor concept is that analog wafer test data can be exploited instead of using pass/fail or bin wafermaps. Hence, deviations can be detected at an earlier stage, before they lead to yield loss and thus, influences costs and quality issues for the manufacturer. Due to the long processing times and restricted possibilities of functional testing during semiconductor frontend production, this aspect is crucial in order to achieve an efficient production.

The aspect of using analog data further deviates from other research works in industry or academia: most of these works extract their information from aggregated data formats, such as bin wafermaps. Furthermore, the Health Factor presented in this work exploits spatial information of the wafermaps to perform pattern recognition, rather than modeling the data distribution in an independent manner (such as e.g. one-dimensional process control systems).

Flexibility Another benefit is that the system permits a flexible, application-specific exchange of the components, which allows transfer to similar problems, e.g. in other industries. For instance, if a process monitoring system for time series data is required, the feature extraction step for the pattern type classifier, as well as the intensity features must be adapted, but the overall concept is still valid for this new objective. This flexibility is a strong benefit of the Health Factor concept, which underlines its value for industrial systems.

Furthermore, the Health Factor is able to handle different product types and technologies within semiconductor industry without major adaptions, provided that a sufficient amount of reliable

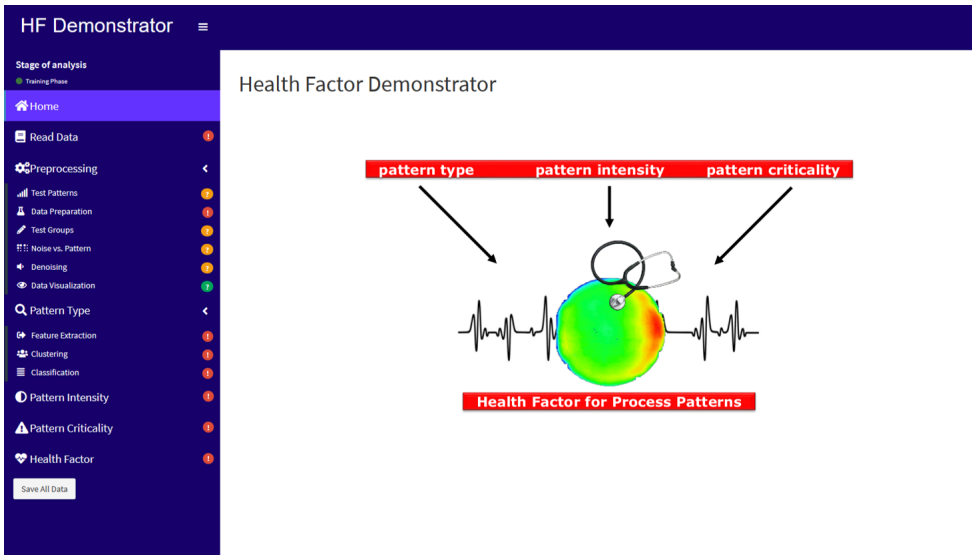


Fig. 4.7: The dashboard presenting the Health Factor, developed R-Shiny [135] in the course of the follow-up EU project ArrowheadTools [12].

training data is available. In particular, the pattern type classifier and the intensity quantifier can be trained on analog wafer test data of any product, unless e.g. the number of elements per wafer is extremely low. In the latter case, it might be necessary to adapt the feature extraction step, accordingly.

Simplicity Upon the aspect of flexibility, the Health Factor benefits from its simple, but sophisticated mathematical structure - as shown in Section 3.4.3, probabilistic as well as deterministic assumptions can be taken into account for the Health Factor. Hence, users are not required to obtain an in-depth understanding of the concept and the technical details to interpret and track the results of the indicator value, since e.g. pattern type class probabilities are easily available.

Another aspect is that the Health Factor can be monitored on different aggregation levels, i.e. the expert can decide the level of information he requires, provided that an appropriate dataset is available. Thus, either a high-level tracking at the level of production lots, or even a wafermap-refined monitoring of e.g. specific test patterns can be provided. As a result, the Health Factor concept supports the expert not only with information obtained from the indicator value, but rather with a hint on possible root-causes behind, i.e. by denoting, which pattern type contributed most to the obtained result. However, results obtained from higher aggregation levels require further evaluation with regard to the usability and the specific requirements in application, which are beyond the scope of this work.

Visualization Apart from the compact mathematical representation of the Health Factor in terms of a real-valued key number, a visualization of the Health Factor concept is developed in the course of the follow-up EU project [12]. Visualization and interactive interfaces are essential to guarantee that users, which are not familiar with the detailed mathematical concept, are able to apply and interpret data-driven key values accordingly. Furthermore, the design of a decision-support system for experts will influence the readiness to apply the system in practice.

In Fig. 4.7, the prototype R-Shiny [135] HTML dashboard for the Health Factor is depicted. In detail, two distinct dashboards account for both, the training and the evaluation phase of the model. Usability for users, such as product experts and operators is a key requirement for the developed tool.

Degree of development Although the interactive dashboard for the Health Factor and the evaluations of the concept demonstrated in the course of this thesis suggest that significant effort was made to underline the performance and applicability of the concept, it is nevertheless necessary to conduct more advanced use-case studies of specific semiconductor products to achieve a higher degree of maturity. The current Technology Readiness Level (TRL) according to the EU Horizon 2020 definition, explained e.g. by Héder [136], is 6, i.e. the validation of the system in specific use-cases. The next stage, i.e. TRL 7 requires the integration of the system in a productive environment and thus, it must be assured that misleading results are avoided and erroneous outputs (e.g. misclassifications) are detected for more diverse datasets. Furthermore, the implementation is to be refined, since all evaluations were done in a development environment in the statistical programming language R [132], while integration into a manufacturing system would require more optimized solutions, as well as interfaces to the existing software landscape. These topics will be a main target of follow-up projects.

Runtime Runtime is a crucial aspect when data-driven tools are concerned. In general, the runtime of the system is dependent on different aspects, including the choice of the parameters (especially for feature extraction), the number of devices per wafer (resolution of the wafermap) and the size of the dataset. Furthermore, the software environment and implementation must not be neglected. All implementations in this thesis were performed in the statistical programming language R [132] and evaluated on a standard hardware (Intel Core i5 processor with 2.30 GHz using a 64-bit Windows 10 operating system). Note that computational overhead might affect the absolute runtimes provided in the following to a limited degree.

With regard to the runtime of the system, the single components must be distinguished: pattern type and pattern intensity comprise most of the computational effort, since setting the criticality values and evaluating the Health Factor formula have minor impact. Upon the preprocessing steps, MRF requires the highest number of elementary operations since it must be calculated for each wafermap separately. However, according to the explicit solution of the system investigated in Section 3.1.5, it is possible to exploit the constant structure of the wafermaps in the Cholesky decomposition, evaluating each wafermap with negligible effort. The most costly part of the pattern type classifier is represented by the feature extraction procedure, depending on the parameter setting (especially the parameters of HOG have a high impact). After reducing the data dimension by feature extraction, subsequent operations (evaluating the machine learning system) have no significant impact on the runtime (approx. 5 seconds for 1000 wafermaps in training or test), unless the number of classes exceeds a reasonable range. However, in this case, quality issues will be likely as well.

Another resource-intense part is the calculation of the autocorrelation measures for intensity quantification. Especially Spatial entropy suffers from a high number of elementary operations when calculating intra- and extra-distances. Since the quality of the results of Spatial entropy

method	runtime
read data	55 sec
TePEx	8 min 35 sec
preprocessing*	29 min 23 sec
MRF	1 min 17 sec
LBP / RLBP	16 min 17 sec
HOG	54 min 12 sec
Naive Bayes	<1 sec
Moran's I	55 min 46 sec
Spatial entropy	1 h 27 min 30 sec
total	≈ 4 h 13 min 56 sec

Tab. 4.14: Runtime results for single evaluation steps of the Health Factor, evaluated for a lot with 1029 wafermaps. * indicates that the preprocessing includes data preparation and format changes, as well as outlier removal, missing value imputation and normalization, but does not contain the MRF and the TePEx procedures.

are additionally observed to be inaccurate in many cases, it is possible to remove Spatial entropy from the default setting if runtime-efficiency is pursued.

Overall, the runtime for the single steps using a lot with slightly more than 1000 wafermaps achieved the runtime results depicted in Tab. 4.14. However, note that these values can be further improved by optimizing the implementations and using programming languages with lower computational overhead.

4.6.2 Aspects of the Pattern Type Classifier

A core part of the Health Factor concept is the pattern type classifier, which has a major influence on the final result of the indicator. However, the pattern type classifier has the highest requirements w.r.t. information provided during training. In order to obtain suitable results of the Health Factor, it is crucial to invest an according effort to provide appropriate training data.

Experimental performance From the experiments, it is obvious that the pattern type classifier is most intensely evaluated upon the three components of the Health Factor. Concerning the performance, it can be shown that the suggested pipeline of preprocessing, feature extraction and supervised machine learning is very successful for the provided datasets. Since most of the compared methods achieved accurate results, the information to discriminate the classes is, obviously, covered by the features and represented in a compact structure.

However, if the types of patterns or the product specifications change fundamentally, it might be necessary to carry out the evaluations, again. In general, a benefit of the procedure is that the parameters of the single steps, e.g. MRF or HOG, can still be modified and thus have the flexibility to easily adapt to new data characteristics.

Concerning single misclassifications, which are contained in the dataset, it is important to keep them in a reasonable range to minimize the probability to miss a wafer, which was subject to

critical deviations, as well as to trigger fail alerts. The achieved accuracy in the experiments showed that such cases, however, cannot be fully avoided for real-world datasets.

Training data & overfitting In general, no machine learning system can perform well, if the information from the training data is insufficient or incorrect - therefore, the amount and quality of training data is crucial for the quality of the system. Although the presented approach is designed to reduce the need for large labeled training datasets (e.g. by applying shallow learning concepts instead of deep learning), the system cannot operate without labeled training data of a specific product. Hence, the expert needs to provide a labeled set of data, grouped by training class in order to train the pattern type classifier.

Concretely, the minimum number of training samples cannot be easily specified. Statistically, the minimum number of training samples corresponds to the number of model parameters - however, the range of this lower bound is by far too low, since the risk of overfitting the data is extremely high if the number of training data is too low compared to the model complexity. Concretely, the requirements of the training sample size will be set by the dimension and the complexity of the chosen machine learning algorithm, which is in turn dependent on several main factors:

- number of classes - in case of a higher number of classes, the multiclass classification problem becomes more *challenging*, i.e. a decision has to be taken among a higher number of options,
- similarity of classes - in case of classes, which show similarities in the feature space, a higher number of training samples is required, unless the parameters of the feature space can be adapted to achieve a better discrimination between the classes,
- intra-class variations - similarly, the number of training samples needs to be larger, if the variations within one or more classes are high,
- existence of new classes - finally, in case that new classes have to be considered by the model (see Section 3.3.3), higher requirements need to be taken into account w.r.t. the volume of training data.

With regard to training data, two possible failure modes can occur: too few labeled training data as well as low-quality training data. In the first case, the well-known problem of overfitting will occur: hence, the machine learning algorithm rather models the characteristics of the single training instances than the underlying class. In fact, the overfitting problem originates from a bad balance between the model complexity and the training sample size. In our case, this imbalance originates from a higher model complexity in combination with too few training data. Hence, the model cannot discriminate between effects from the class model (which should be modeled) and effects from single instances (which should not be modeled) and achieves a bad predictive quality.

On the other hand, it may happen that a sufficient amount of training data is available, but the quality of this dataset is low, since e.g. it shows an erroneous labeling or does not cover the full variety of certain classes. In both cases, prediction quality of the machine learning method will decline significantly.

Underfitting The opposite problem of overfitting is also relevant for the Health Factor: in case that the model complexity (indicated by the number of model parameters) is too low, it might not be possible to cover the variations within the classes. This issue is a major restriction of the suggested methods, in case that the system is transferred to a product, which shows more complex pattern types. In this case, it is possible that the assumptions taken by the classifier are not sufficient to discriminate between the classes correctly. One example would be that a linear classifier will not be able to correctly distinguish between two classes, which are separated by a quadratic function. In addition, an inappropriate selection of the features can result in underfitting.

If the problem originates from the classifier, two possible solutions exist: on the one hand, it is possible to apply a more flexible classifier at the cost of requiring more training data. On the other hand, it is possible in many cases to transform the data into an adapted feature space, where the classes are linearly separable (a similar concept to the kernel trick for SVMs). Hence, adapting the feature space can resolve this issue.

Unknown classes Another aspect, which must be discussed regarding the pattern type classifier is the suggested extension towards datasets with new classes. Due to the lack of available training data, a semi-supervised setting has to be chosen to detect new classes. Hence, the most value can be obtained from the combination of both, labeled and unlabeled data, which is a promising approach for the semiconductor industry, where unlabeled data are available at a large volume for many products.

The proposed method for semi-supervised classification with unknown classes combines sophisticated statistical frameworks, i.e. the Bayesian paradigm with the GMM model and hence, provides a generative model. Accordingly, information about the data distribution is gained indirectly from the model parameters. Moreover, the BIC criterion applied for estimating the number of GMM components provides a vote for the number of unknown classes provided in the dataset (although tends to overestimate this number).

On the other hand, consider the case that data are more difficult to separate within the feature space (especially in high-dimensional features spaces): for this setup, discriminative methods can achieve equal or even better results, see Schrunner et al. [17]. Furthermore, the achieved prediction quality of the classes is not yet in the range of the supervised results with purely known classes, as can be seen from experiment 1. The problem setup is, indeed, significantly more complex to the supervised classification problem. Hence, elaborating on the method to achieve more appropriate results will be beneficial for using this methodology in practical setups.

4.6.3 Aspects of the Pattern Intensity Quantifier

In contrast to the pattern type classifier, the pattern intensity quantification step is purely based on unsupervised settings, since labeled data is even more difficult to generate. Hence, the evaluation of this step is difficult and could only be performed using simulated data. The pattern intensity contributes to the Health Factor independently from the pattern type and enables the expert to automatically judge the degree of development of a pattern.

Novelty A major aspect of the intensity quantification is its novelty: the concept was not yet, up to the knowledge of the author, described in literature for a comparable case, apart from the publications related to this work. Hence, the choice of baseline methods or evaluation setups is also limited.

However, the idea to quantify the stage of development of a pattern between weak and strong ones is reasonable when considering the overall concept to develop an early detection system for process deviations. In particular, it might be essential for the experts to get this type of information to avoid that the number of triggered alerts is too high in an operating manufacturing system due to many weakly developed deviations, which are not associated with quality concerns.

Methodology The proposed methods, i.e. Moran's I and Spatial entropy proved to be useful for the intended task. Nevertheless, the adaption using image segmentation in order to restrict these methods to smaller ROIs is necessary, since background structures massively distort the intensity quantification procedure.

In contrast to Moran's I, Spatial entropy suffers from large computational effort when calculating intra and extra distances, which is a drawback with regard to runtime. Furthermore, the results show that Moran's I achieved slightly better results as a stand-alone method. The combination of both methods via dimensionality reduction or feature selection appears to be helpful for more complex datasets, where the characteristics of pattern types are very different from each other.

The evaluation of pattern intensity in an unsupervised way is inevitable, since labeled training data cannot be provided reliably for real-world products. However, unsupervised learning is associated with the drawback that the user cannot influence the choice of the method for intensity quantification, but has to rely on a suitable choice taken by the machine. It may happen that the most significant variation depicted numerically within the feature set is not equivalent to the optically salient feature of intensity. Hence, the method could provide bad results due to spurious variations in the data.

Absolute range of intensity Another limitation of the current intensity concept is that the intensity is bounded between 0 and 1 (by definition), but the actual data range is not further specified. Hence, e.g. Moran's I delivers values between 0.8 and 0.99 for most wafermaps across all patterns in the real-world datasets.

However, if another type of intensity features is extracted from the dataset, it is possible that the relative data range is larger or shifted within the interval $[0, 1]$. Hence, comparability between the single methods must be guaranteed, e.g. by appropriate normalization. Another solution would be to specify the intensity by a single method, independently from the pattern type. However, results showed that the intensity quantification methods yield results of unequal quality when applying stand-alone methods.

4.6.4 Aspects of the Pattern Type Criticality

The third component of the Health Factor, the pattern type criticality, mainly introduces expert knowledge to the system. A replacement by summary statistics of historical data (e.g. yield loss) would be conceivable at first glance, but since the Health Factor quantifies patterns within the specification limits, the yield loss of such samples is hardly representative. The specification, i.e.

freely assigning a value between 0 and 1, can be judged as a benefit or drawback - one argument is that it enables flexibility for the expert, another would be that the expert has to know and understand the root-cause of the pattern in detail to be able to specify the value.

Flexibility Firstly, assessing the manual assignment from an optimistic perspective, the expert is able to select a specific focus of the data evaluation by assigning high criticality values to the pattern types of interest without having to consider the full set of possible patterns. Hence, the Health Factor is applicable for both, product experts with interest on all patterns affecting their product, as well as process experts with interest in the subset of patterns related to their maintained processes.

Furthermore, the granularity of the criticality values helps experts to adapt the procedure to a specific product. By assigning distinct levels to the single pattern types, it is possible to track large datasets of a specific product with large numbers of distinct pattern types at distinct criticality levels by a single Health Factor key value. On the other hand, if few information is available for a specific product, the expert can resume to the binary $0-1$-setting for the criticality values.

Finally, the setting of the criticality values can be changed without having to recalculate or retrain the other components, i.e. an erroneous assignment of the criticality values can be corrected without large effort in practice.

Understanding root-causes Detecting a root-cause in semiconductor industry is a difficult task and hence, the quantification of the associated criticality on a $0-1$-scale is difficult. Complex physical relations and specifications of the product must be taken into account for this reason.

However, a lot of historical root-cause analyses exist for semiconductor products, which were subject to serious yield loss or customer claims in their history. Hence, a lot of knowledge is available from product experts. Furthermore, it holds in many cases that the longer a product type has been in mass production, the more deviations and hence, the more root-causes for process patterns are known. As a result, specifying the criticality value with a reasonable precision will be heavily influenced by the age of the product under investigation.

Semi-automated systems In order to quantify whether a pattern type has criticality 0.5 or 0.51 will be difficult based on expert judgment. Thus, different heuristics and semi-automated systems to choose the criticality values are proposed in Section 3.4.2. However, there might exist many other systems, which are intended to propose root-causes to available wafer test data. Furthermore, the development of such systems is a topic of ongoing research, such as e.g. in [11].

4.6.5 The Validity of Fundamental Assumptions

Finally, we resume to the assumptions taken within this work and analyze their validity. The two major assumptions stated in the introduction were Assumption 1 and Assumption 2 (Section 2.3.1), i.e. that root-causes of patterns are identifyable and that only one pattern is present on each wafermap. In general, both assumptions are required to obtain a system with a manageable complexity.

Identifyability From a different perspective, identifyability means whether or not we can be sure that two distinct root-causes show two distinct patterns. If, however, two process steps deliver the same pattern type, it is very likely that these pattern types are detected in vastly different electrical parameters during wafer test. Hence, the expert will be able to discriminate between these root-causes by taking this meta-information of the wafermap into account.

The wafer test sequence is designed such that different types of functional errors can be captured. Consequently, wafermaps from the vastly different electrical parameters should capture almost all types of critical process deviations. However, some electrical tests might be redundant or highly correlated, i.e. it is possible that the same pattern occurs in more than one wafermap of the same wafer. Nevertheless, this aspect is unaffected from the identifyability assumption.

Uniqueness In real-world datasets, mixtures of distinct patterns occur frequently, but in most cases, a main pattern of interest is optically salient. Hence, additional patterns contribute to the background structures evaluated e.g. for intensity quantification. Furthermore, the smoothing procedure by MRFs removes not only measurement noise, but also reduces the effects of such background.

For the (less likely) case that two patterns are demixed with similar intensities, root-causes of both patterns will be similar - otherwise, they would not be covered by the same electrical parameter during wafer test. Hence, the expert may also want to judge the combined pattern as a new pattern type and assign it to a separate criticality value.

Furthermore, application of demixing strategies, e.g. by blind source separation techniques such as ICA, were discussed in Section 2.3.1.

5 Conclusion

5.1 Resume

In summary, this work tackles the automated analysis of wafer test data by modern pattern recognition techniques for industry 4.0 applications in semiconductor manufacturing. More concretely, we defined a Health Factor, which can be interpreted as an indicator for the presence of critical process deviations, depicted indirectly via process patterns in electrical measurement data. We stated two conditions, which are necessary for the success of this system, i.e. that each root-cause is identifyable given the wafer test data, as well as that only one pattern is present on the wafermap. We discussed the validity of these assumptions in Section 4.6.

We first explained the structure and background of wafer test data, as well as the problem setup. In course of that, we highlighted that wafer test data are affected by many erroneous influences, such as measurement noise, regularities from the testing procedure (test patterns), outliers and missing values. In order to cope with these characteristics, we observed that a sophisticated preprocessing procedure is essential to extract knowledge from the data. Hence, preprocessing was investigated as a first part of Section 3.

We emphasized the value of the present work by providing a review on related work in semiconductor industry, where several aspects deviated from our investigated setting: firstly, most works were based on pass/fail or bin instead of analog wafermaps, which contain less information on the process deviation. Secondly, most works require large amounts of training data, neglect variations within the process pattern types (e.g. rotation) or have a purely academic focus. As a third aspect, most literature works cannot integrate expert knowledge in the system, since they exclusively focus on pattern recognition, which is a major difference to the presented setup.

In order to reach the objective to define a Health Factor, a system containing three distinct components was proposed: the pattern type classifier, the intensity quantifier and the pattern criticality. While the first two components are delivered by data-driven systems, the latter is provided by the expert, who judges the severity of known pattern types on a binary scale.

A core part of the system was the pattern type classifier, i.e. the component, which is responsible for performing pattern recognition within the set of known process patterns. From a technical perspective, preprocessing via a MRF model was suggested for this purpose, followed by an engineering-based feature extraction approach via LBP, RLBP and HOG, and a selection of suitable classifiers from supervised machine learning. The setting was evaluated in the experiments presented in Section 4.3, where it provided accurate results for different types of classification methods.

Furthermore, we suggested an extension of the supervised setting: in order to additionally classify new instances into one of the known classes (multiclass classification) or into a new class, it was necessary to integrate more information than contained in the labeled training data. For this purpose, a semi-supervised classifier for datasets with unknown classes was proposed and evaluated. This method follows a generative Bayesian classification paradigm and extends the concept by integrating a residual risk that the instance belongs to none of the known classes.

After recognizing the category of the pattern, each pattern type has to be assigned a so-called criticality value, i.e. a number denoting how critical the process deviation associated to the respective pattern type is. This value is commonly assigned by experts, but can also be chosen by heuristics in an automated or semi-automated way, exploiting information from additional data sources. In Section 4.6, we reviewed the different viewpoints on whether this free choice of the criticality values should be rather considered a benefit or a limitation of the system.

Independent from the pattern type and its criticality, the pattern intensity is a concept, which describes an intrinsic characteristic of the wafermap: the idea is that a process pattern on the wafermap has different degrees of development, which can be distinguished. In practice, a wafermap showing a pattern type with high intensity (i.e. a strong pattern) is associated with a higher Health Factor value than a wafermap, where the same pattern type is represented with lower intensity. Semantically, intensity is used to describe the strength of a process deviation. With regard to the methodology, the intensity concept is modeled via statistical autocorrelation methods, which judge the degree of mutual correlation between devices in a spatial neighborhood on the wafer.

After investigating the three components in detail, the final Health Factor was derived using a concept, which originates from statistical decision theory. Therefore, a loss function was defined, denoting the level of concern, which should be assigned if a certain state of nature (i.e. a certain pattern type) occurs. The concept was extended for both, probabilistic and deterministic choices of the intensity and criticality components, while the pattern type was always modeled in a probabilistic way. Although originally defined for the lowest aggregation level, i.e. a single wafermap, the concept was elevated to higher aggregation levels, such as wafer or lot level.

The experimental evaluation of the concept was divided into two parts: first, the single components were investigated, then the final Health Factor was validated. The experiments were conducted on 3 datasets, comprising one dataset with simulated, online available data and 2 real-world datasets from a semiconductor manufacturer. The datasets comprised between 4 and 8 distinct pattern types.

The evaluation of the single components demonstrated that the pattern type classifier achieves good performance, which is obviously related to an accurate choice of image features (since all different machine learning methods provide good results). Furthermore, also the intensity quantifier performed well, although fewer experiments could be made due to the limited availability of reliable ground truth values or rankings.

In the validation step of the Health Factor concept, we observed that the concept provides interpretable and reasonable results, which is a necessary condition for deployment in real-world environments. Most of the wafermaps, which showed a strongly developed, critical pattern in the ground truth, could be recognized by high Health Factor values. The number of false alarms (false positives) was kept in a reasonable range. The runtime evaluations showed that the processing

time for one wafermap was in the range of seconds, whereas a full production lot took several hours (depending on the number of devices per wafer and the processing hardware).

Finally, in Section 4.6, we discussed the single components, as well as the overall concept, highlighting the benefits and drawbacks of the proposed approaches. In particular, additional aspects, such as flexibility and runtime, were taken into account at this stage.

5.2 Key Learnings

Finally, the major conclusions from the investigations accompanying this work shall be reviewed. Upon all aspects mentioned in the course of this thesis, a lot of observations were made when experimenting with wafer test data. Here, key learnings are summarized, which partially led to a change in the intended approach to achieve the objectives.

Why we need expert knowledge An initial idea to approach the Health Factor was to link the process pattern directly to the process deviation. However, one major observation was that the wafermap data carries the information on the pattern type and the intensity, but not the information, where root-cause can be found in the process, unless additional, extrinsic sources of information are provided. Since this study is purely based on wafer test data, the only possibility to integrate this process link is to exploit expert knowledge. More concretely, the process link was integrated via the pattern criticality in an indirect way.

In particular, the need for expert knowledge influenced two aspects of the Health Factor concept: firstly, the pattern criticality value was introduced to insert extrinsic information to the data-driven pipeline. Secondly, the decision on the present pattern type was taken based on supervised or semi-supervised, instead of unsupervised techniques.

Why clustering is not suitable for pattern type recognition A related aspect is the choice of a supervised approach for the pattern type classifier: the idea of a fully-automated system collides with the concept of supervised learning, since (at least for training), a high effort is demanded from the expert to provide accurate training data. Hence, automation is often associated with unsupervised learning. For the pattern type, however, clustering yielded several limitations in contrast to classification.

Since each type of pattern shows distinct characteristics, e.g. rotation- or position-invariance, distinct sets of features are required to discriminate between pattern types. However, a clustering approach, which is purely based on a distance measure cannot account for these characteristics of each class, since it treats all features equally. Hence, it will take decisions based on intrinsic correlations. As a result, the clustering result will hardly coincide with the expected result from human classification.

Furthermore, clustering will intrinsically form groups, which cannot be easily associated with the extrinsic expert information on the process link or process criticality. However, as explained above, the criticality information on the pattern types, which is provided by the expert is also essential for the Health Factor. For clustering results, the criticality values for each pattern type have to be collected after obtaining the clusters, which is not reasonable for a (mostly) automated system.

Finally, the number of clusters is a parameter, which has to be chosen by an expert (or by a heuristic, which does not represent reality in many situations) and hence, the unsupervised clustering approach is, again, subject to expert interference.

Importance of interpretable results In an industrial, real-world environment, it is crucial to provide decisions, which are comprehensive and interpretable for an expert. For this purpose, it is important to deploy statistical frameworks, which can provide explicit probability models for the according decision - generative approaches are favored, if they achieve a similar quality as discriminative ones. In case that a decision is not understood, it will most likely not be trusted, especially in case that the economic impact is high. As a result, the system might be overruled by a human expert. Hence, it is important to convey confidence by the suggested automation system. A major contribution to this is to provide an explanation on the mode how the suggested decision is generated, which contradicts the black-box model of deep learning systems. Although a generative model does not guarantee full interpretability in general, it is associated with a higher exploratory power.

In the present work, we emphasized the importance of this aspect by deploying sophisticated, well-established mathematical and statistical frameworks instead of black-box models. In particular, it is highly important to investigate the statistical properties of a system from a theoretic, as well as from an experimental perspective.

Transferability of single components Apart from the main objective of this research work, several by-products were created, i.e. some of the components are transferable to different applications within semiconductor industry. For example, the TePEx procedure can be applied as a stand-alone solution to observe the quality of the testing procedure of semiconductor products.

Furthermore, pattern recognition might not only be of high importance to the Health Factor for process monitoring, but also applicable e.g. for quality control of products, either embedded in a framework or as a stand-alone system. Intensity quantification promises to be valuable for quality applications, as well.

Finally, the Health Factor concept can be deployed to construct any other indicator related to process control or monitoring, which is based on automated or semi-automated data analysis. For this purpose, distinct choices for pattern type recognition or intensity might be necessary - accounting for different data types or characteristics, such as e.g. time series data. One possible application could be e.g. predicted maintenance of manufacturing equipment.

Demonstration of data science in manufacturing An important aspect of this research work is, finally, to demonstrate that automated or semi-automated systems can be successfully applied in manufacturing environments to relieve the expert from monitoring tasks and bundle the attention to the few failure cases, which might occur.

Although industry 4.0 has been a topic of high interest in the recent years, suspiciousness and lack of knowledge about the possibilities and limitations are still present in industrial environments. Hence, it is essential to demonstrate and integrate automated systems into manufacturing, which provide a high performance and can be used to convince experts from their benefits.

5.3 Self Reflection

In conclusion of this PhD thesis, a self-reflection of the major benefits and limitations related to the presented concept as well as the scientific and industrial contribution shall be provided. The following section is based on a subjective evaluation of the goals achieved within this work, but also points out specific limitations.

The major goal of the Health Factor was to accomplish a decision support system for experts to judge the degree, to which a wafer is affected by critical process patterns. In general, this objective could be met, as the suggested Health Factor provides a compact representation to monitor this information, given that the components fulfill their subtasks correctly. Since in addition, the concepts of intensity and criticality were introduced as novel aspects, the system can be considered as a large step from the industrial perspective.

Furthermore, the suggested Health Factor shows the benefit that it is not restricted to specific products, production processes or pattern types. The concept is rather flexible and can be transferred to other device types. In addition, the components can be easily exchanged in order to tailor the setting to new requirements.

However, upon defining a mathematical construct, the required evaluation is still to be consolidated. Since no evaluation metric was able to objectively track and extensively verify the information content of the Health Factor, it is still to be shown in practical applications, to what degree the system is able to provide the intended decision support or needs to be adapted.

Concerning the suggested Health Factor components, the pattern type classifier comprised a large part of this thesis, achieving good results in the experimental evaluation. However, the datasets may not represent the full variety of semiconductor production, thus not only more evaluation might be needed in order to construct a productive system, but also the features extracted to cover the pattern type characteristics might not be sufficient in case that previously unseen pattern types occur.

The same is true for the intensity quantifier, where the suggested autocorrelation concepts might require more intense evaluation in practical case studies. Furthermore, also the intensity concept might be required to comprise more distinct concepts in order to cover new pattern types. In general, a joint intensity quantification method for all pattern types would be beneficial to assure comparability of the Health Factor values between instances from different pattern types, but this goal is hard to achieve due to distinct characteristics of the patterns.

With regard to the scientific contribution, the focus of this work was driven by the needs to solve the industrial problem rather than by method-driven research interests. From this perspective, the objective was accomplished, since the industrial problem could be resolved accordingly. In particular, the development of the semi-supervised classifier with unknown classes represented a key development, which has value for the machine learning community and is further transferable to related problem setups from other applications.

On the other hand, the field of deep learning, which currently attracts large interest in the data science community, mainly remained untouched by the presented research. Although deep learning methods might be of research interest, the problem setup (few labeled data, no publicly available large datasets, etc.) was restrictive in this respect. In addition, black-box models are exposed to skepticism of users.

Overall, the presented thesis represents a synergy between industrial and academic objectives, which is considered the most significant strength of this work: sophisticated methods are adapted and developed in order to solve a complex real-world problem. In contrast to many data science projects, the research questions raised the necessity to extensively understand and exploit both, information extracted from data and domain knowledge from semiconductor industry. The suggested Health Factor meets these requirements and contributes to both, industry and academia.

5.4 Outlook

Following this research project, multiple follow-up activities within related EU-projects are intended to expand the Health Factor system and its components. Both, industrial as well as academic topics within this work can be exploited in future work.

Firstly, the EU-project iDev40 [11] approaches the problem of detecting process deviations from an advanced perspective: instead of judging the quality of the process via wafer test data, it is intended to construct a link to the production process by taking additional data sources into account. One possibility is to analyze machine parameters of the manufacturing equipment, so-called APC data. Furthermore, results from inline measurements of various types (e.g. optical control, thickness tests, etc.) can be exploited, as well as logistical information from the production will be considered. However, the diversity of data sources raises huge issues with data consistency and requires flexible methods to detect correlations within this big data system.

Furthermore, the EU-project Arrowhead Tools [12] is intended to continue the development of the proposed Health Factor (as well as other data-driven approaches) towards integration into a real-world production environment. In detail, the output of the present work is a demonstrator, which has to undergo further evaluations, consistency checks, as well as optimization procedures. In addition, the implementation needs to be adapted to the IT environment available in production and an optimized exploitation of the information considering the practical demands is required. Finally, the experts need to be made familiar with the system and its usage.

Upon these aspects, a clear goal in semiconductor industry is to develop systems, which include both, frontend and backend production. Using such systems, it is possbile to track errors between frontend and backend, otherwise undetected errors originating from frontend are hardly detectable in backend production. Such a system, however, needs to cope with the different logistical aggregations in frontend and backend (i.e. the wafer vs. the single-chip processes), and require that e.g. single devices can be traced back to their wafer position during backend. This requirement is not necessarily fulfilled for all products.

From a scientific perspective, especially the problem of taking unknown classes into account is of major interest. Although the present work suggests an algorithm to cope with this requirement, it is still related to a significantly larger need for training data, as well as it leads to a lower accuracy compared to multiclass classifiers with purely known classes. Hence, more research in the direction of open set recognition, exploratory learning and PU-learning using real-world datasets is necessary.

Finally, another machine learning paradigm, which was not part of this work, would be of major interest for future work: transfer learning. Using transfer learning approaches, it is possible to exploit the information from machine learning systems, which were trained using a similar

dataset or for a similar purpose. Using this information, the need for labeled training data is minimized compared to training a new machine learning system from scratch. Especially for deep learning systems, transfer learning provides accurate results. However, for semiconductor industry, the limited availability of open data is a restriction for the development of deep learning systems, which require large amounts of training data and could be used for transfer learning.

Nevertheless, transfer learning systems could be based on simulated wafer test data, such as those provided online in the course of this thesis. Hence, transfer learning systems are a promising way to overcome the limitations caused by the availability of real-world data from this industry.

List of Figures

List of Tables

Acronyms

APC Advanced Process Control

ASC Average Silhouette Coefficient

ATE Automatic Test Equipment

BEOL Back End of Line

BI Burn-In

BIC Bayesian Information Criterion

CNN Convolutional Neural Network

CVAE Convolutional Variational Autoencoder

DSS Decision Support System

ECOC Error-correcting output coding

EM Expectation Maximization

FEOL Front End of Line

GaN Gallium Nitride

GMM Gaussian Mixture Model

HBIN Hard Bin

Health Factor Health Factor for Process Patterns

HOG Histogram of Oriented Gradients

I-EM Initial EM

ICA Independent Component Analysis

IGBT Integrated-Gate Bipolar Transistor

iid independent and identically distributed

IQR Inter-Quartile Range

IRDS International Roadmap for Devices and Systems

ITRS International Technology Roadmap for Semiconductors

LBP Local Binary Pattern

MAR missing at random

MCAR missing completely at random

MICE Multivariate Imputation by Chained Equation

MRF Markov Random Field

NMAR not missing at random

NMF Non-negative Matrix Factorization

NMI Normalized Mutual Information

NNR Nearest Neighbor Residuals

PAT Part Average Testing

PCA Principal Component Analysis

PCM Process Control Monitoring

RLBP Rotated Local Binary Pattern

ROC Receiver operating characteristic

ROI Region of Interest

RTP Rapid Thermal Processing

S-EM Spy EM

SBIN Soft Bin

SiC Silicon Carbide

SIFT Scale-Invariant Feature Transform

SSC-UC Semi-Supervised Classifier with Unknown Classes

SVM Support Vector Machine

TePEx Test Pattern Extraction

TRL Technology Readiness Level

Bibliography

[1] M. Skilton and F. Hovsepian, *The 4th Industrial Revolution: Responding to the Impact of Artificial Intelligence on Business*. Cham, Switzerland: Palgrave Macmillan, 2018.

[2] US National Science Foundation, Directorate for Engineering, "Cyber-physical systems (cps)." https://www.nsf.gov/pubs/2010/nsf10515/nsf10515.htm, 2010. [Online; accessed August 22, 2019].

[3] F. Mattern and C. Floerkemeier, *From the Internet of Computers to the Internet of Things*, pp. 242–259. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

[4] C. Roser, "Illustration of industry 4.0." https://upload.wikimedia.org/wikipedia/commons/c/c8/Industry_4.0.png, Nov 2015. [Online; accessed April 08, 2019].

[5] WSTS Inc., "World semiconductor trade statistics." https://www.wsts.org, 2019. [Online; accessed April 10, 2019].

[6] Statista Inc., "Semiconductor sales revenue worldwide from 1987 to 2019." https://www.statista.com/statistics/266973/global-semiconductor-sales-since-1988, 2019. [Online; accessed April 10, 2019].

[7] D. Ferrucci, A. Levas, S. Bagchi, D. Gondek, and E. T. Mueller, "Watson: Beyond jeopardy!," *Artificial Intelligence*, vol. 199-200, p. 93–105, Aug 2012.

[8] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, p. 484–489, Jan 2016.

[9] "Power semiconductor and electronics manufacturing 4.0." http://www.semi40.eu. [Online; accessed April 09, 2019].

[10] Infineon Technologies AG, "Wafer overview." https://www.infineon-brandportal.com/media-pool/asset/directDownload/2221/lowres, Aug 2014. [Online; accessed April 10, 2019].

[11] "Integrated development 4.0." http://www.idev40.eu. [Online; accessed April 09, 2019].

[12] "Arrowhead framework." https://arrowhead.eu/arrowheadtools. [Online; accessed April 09, 2019].

[13] S. Schrunner, O. Bluder, A. Zernig, A. Kaestner, and R. Kern, "Markov random fields for pattern extraction in analog wafer test data," in *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pp. 1–6, Nov 2017.

[14] M. Pleschberger, S. Schrunner, and J. Pilz, "An explicit solution for image restoration using markov random fields," *Journal of Signal Processing Systems*, Aug 2019.

[15] T. Santos, S. Schrunner, B. C. Geiger, O. Pfeiler, A. Zernig, A. Kaestner, and R. Kern, "Feature extraction from analog wafermaps: A comparison of classical image processing and a deep generative model," *IEEE Transactions on Semiconductor Manufacturing*, vol. 32, pp. 190–198, May 2019.

[16] S. Schrunner, O. Bluder, A. Zernig, A. Kaestner, and R. Kern, "A comparison of supervised approaches for process pattern recognition in analog semiconductor wafer test data," in *17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 820–823, Dec 2018.

[17] S. Schrunner, B. C. Geiger, A. Zernig, and R. Kern, "A generative semi-supervised classifier for datasets with unknown classes," 2019. manuscript under preparation.

[18] S. Schrunner, A. Jenul, M. Scheiber, A. Zernig, A. Kaestner, and R. Kern, "A health factor for process patterns - enhancing semiconductor manufacturing by pattern recognition in analog wafermaps," in *IEEE International Conference on Systems, Man and Cybernetics*, 2019. manuscript accepted for publication.

[19] S. Schrunner, A. Zernig, A. Kaestner, and R. Kern, "Process monitoring in industry 4.0 - a framework for detecting process deviations based on pattern recognition," *Special Section on Data science challenges in Industry 4.0 in IEEE Transactions on Industrial Informatics*, 2019. manuscript submitted for publication.

[20] B. Geiger, S. Schrunner, and R. Kern, "An information-theoretic measure for pattern similarity in analog wafermaps," in *European Advanced Process Control and Manufacturing Conference*, Apr 2019. Conference date: 08-04-2019 through 10-04-2019.

[21] A. Jenul, S. Schrunner, A. Zernig, A. Kaestner, and J. Pilz, "An investigation of statistical measures for intensity comparison of process patterns in analog wafer test data," in *European Advanced Process Control and Manufacturing Conference*, Apr 2019. Conference date: 08-04-2019 through 10-04-2019.

[22] Infineon Technologies AG, "Infineon succeeds in producing chips on new 300-millimeter thin wafer technology for power semiconductors." https://www.infineon.com/cms/en/about-infineon/press/press-releases/2011/INFXX201110-002.html, 2011. [Online; accessed April 10, 2019].

[23] J. Camassel, S. Contreras, and J.-L. Robert, "Sic materials: a semiconductor family for the next century," *Comptes Rendus de l'Académie des Sciences - Series IV - Physics*, vol. 1, pp. 5 – 21, 2000.

[24] T. J. Flack, B. N. Pushpakaran, and S. B. Bayne, "Gan technology for power electronic applications: A review," *Journal of Electronic Materials*, vol. 45, pp. 2673–2682, Jun 2016.

[25] P. Gargini, "The international technology roadmap for semiconductors (itrs): "past, present and future"," in *GaAs IC Symposium. IEEE Gallium Arsenide Integrated Circuits Symposium. 22nd Annual Technical Digest 2000. (Cat. No.00CH37084)*, pp. 3–5, Nov 2000.

[26] IEEE, "International roadmap for devices and systems." https://irds.ieee.org. [Online; accessed April 10, 2019].

[27] STMicroelectronics, "Application note: Introduction to semiconductor technology." https://www.st.com/content/ccc/resource/technical/document/application_note/f1/36/51/95/ff/f3/44/19/CD00003986.pdf/files/CD00003986.pdf/jcr:content/translations/en.CD00003986.pdf, 2000. [Online; accessed April 24, 2019].

[28] A. Klemmt, *Ablaufplanung in der Halbleiter- und Elektronikproduktion*. Wiesbaden, Germany: Vieweg+Teubner Verlag | Springer, 2012.

[29] V. Alagić, "Test pattern extraction for semiconductor wafer test data," Master's thesis, Alpen-Adria-Universität Klagenfurt, Sep 2017.

[30] J. Rivoir, "Parallel test reduces cost of test more effectively than just a cheap tester," in *IEEE/CPMT/SEMI 29th International Electronics Manufacturing Technology Symposium (IEEE Cat. No.04CH37585)*, pp. 263–272, July 2004.

[31] D. Kurz, H. Lewitschnig, and J. Pilz, "Decision-theoretical model for failures which are tackled by countermeasures," *IEEE Transactions on Reliability*, vol. 63, pp. 583–592, June 2014.

[32] D. Kurz, H. Lewitschnig, and J. Pilz, "An advanced area scaling approach for semiconductor burn-in," *Microelectronics Reliability*, vol. 55, pp. 129 – 137, 2015.

[33] A. Zernig, *Device level Maverick Screening*. PhD thesis, Alpen-Adria-Universität Klagenfurt, Jul 2016.

[34] S. Mittal, "Parallel test reduces cost of test more effectively than just a cheap tester," *ACM Computing Surveys,*, vol. 48, pp. 54:1–54:29, May 2016.

[35] Automotive Electronics Council, "Guidelines for statistical yield analysis." http://www.aecouncil.com/Documents/AEC_Q002_Rev_A.pdf, Aug 2012. [Online; accessed April 12, 2019].

[36] Automotive Electronics Council, "Guidelines for part average testing." http://www.aecouncil.com/Documents/AEC_Q001_Rev_D.pdf, Dec 2011. [Online; accessed April 12, 2019].

[37] L. Pötsch, *Early Failure Detection by Robust Statistics and Spatial PointProcesses*. PhD thesis, Alpen-Adria-Universität Klagenfurt, Jun 2010.

[38] W. R. Daasch, K. Cota, and J. McNames, "Neighbor selection forvariance reduction in iddq and other parametric data," in *Proceedings International Test Conference 2001*, pp. 92–100, Nov 2001.

[39] A. Zernig, O. Bluder, J. Pilz, and A. Kaestner, "Device level maverick screening - detection of risk devices through independent component analysis," in *Proceedings of the Winter Simulation Conference 2014*, pp. 2661–2670, Dec 2014.

[40] A. Zernig, O. Bluder, J. Pilz, A. Kaestner, and A. Krauth, "Identification of risk devices using independent component analysis for semiconductor measurement data," *International Journal of Industrial Engineering: Theory, Applications and Practice*, vol. 23, Jan 2017.

[41] R. Turakhia, B. Benware, R. Madge, T. Shannon, and R. Daasch, "Defect screening using independent component analysis on iddq," in *23rd IEEE VLSI Test Symposium (VTS'05)*, pp. 427–432, May 2005.

[42] J. Bartholomäus, S. Wunderlich, and Z. Sasvári, "Identification of suspicious semiconductor devices using independent component analysis with dimensionality reduction," in *ASMC Advanced Semiconductor Manufacturing Conference*, May 2019.

[43] R. Schachtner, *Extensions of non-negative matrix factorization and their application to the analysis of wafer test data*. PhD thesis, Universität Regensburg, Feb 2010.

[44] R. Schachtner, G. Pöppel, A. M. Tomé, and E. W. Lang, "Minimum determinant constraint for non-negative matrix factorization," in *Independent Component Analysis and Signal Separation*, (Berlin, Heidelberg), pp. 106–113, Springer Berlin Heidelberg, 2009.

[45] R. Schachtner, G. Pöppel, and E. W. Lang, "A nonnegative blind source separation model for binary test data," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, pp. 1439–1448, Jul 2010.

[46] R. Schachtner, G. Pöppel, and E. W. Lang, "Towards unique solutions of non-negative matrix factorization problems by a determinant criterion," *Digital Signal Processing*, vol. 21, no. 4, pp. 528 – 534, 2011.

[47] T. Siegert, R. Schachtner, G. Poeppel, and E. W. Lang, "A nonnegative tensor factorization approach for three-dimensional binary wafer-test data," in *15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 842–845, Dec 2016.

[48] G. R. Naik, D. K. Kumar, and M. Palaniswami, "Multi run ica and surface emg based signal processing system for recognising hand gestures," in *8th IEEE International Conference on Computer and Information Technology*, pp. 700–705, July 2008.

[49] J. K. Kibarian and A. Strojwas, "Using spatial information to analyze correlations between test structure data (semiconductor ic manufacture)," *IEEE Transactions on Semiconductor Manufacturing*, vol. 4, pp. 219–225, Aug 1991.

[50] M. W. Cresswell, D. Khera, L. W. Linholm, and C. E. Schuster, "A directed-graph classifier of semiconductor wafer-test patterns," *IEEE Transactions on Semiconductor Manufacturing*, vol. 5, pp. 255–263, Aug 1992.

[51] K. Zinke, M. B. Nasr, A. Hicks, M. Crawford, and R. Zawrotny, "Yield enhancement techniques using neural network pattern detection," in *1997 IEEE/SEMI Advanced Semiconductor Manufacturing Conference and Workshop ASMC 97 Proceedings*, pp. 211–215, Sep. 1997.

[52] C.-J. Huang, C.-F. Wu, and C.-C. Wang, "Image processing techniques for wafer defect cluster identification," *IEEE Design and Test of Computers*, vol. 19, pp. 44–48, Aug 2002.

[53] S.-C. Hsu and C.-F. Chien, "Hybrid data mining approach for pattern extraction from wafer bin map to improve yield in semiconductor manufacturing," *International Journal of Production Economics*, vol. 107, pp. 88–103, May 2007.

[54] F.-L. Chen and S.-F. Liu, "A neural-network approach to recognize defect spatial pattern in semiconductor fabrication," *IEEE Transactions on Semiconductor Manufacturing*, vol. 13, pp. 366–373, Aug 2000.

[55] L.-C. Chao and L.-I. Tong, "Wafer defect pattern recognition by multi-class support vector machines by using a novel defect cluster index," *Expert Systems with Applications*, vol. 36, pp. 10158–10167, Aug 2009.

[56] T.-S. Li and C.-L. Huang, "Defect spatial pattern recognition using a hybrid som–svm approach in semiconductor manufacturing," *Expert Systems with Applications*, vol. 36, pp. 374–385, Jan 2009.

[57] M.-J. Wu, J.-S. R. Jang, and J.-L. Chen, "Wafer map failure pattern recognition and similarity ranking for large-scale data sets," *IEEE Transactions on Semiconductor Manufacturing*, vol. 28, pp. 1–12, Feb 2015.

[58] J. Yu and X. Lu, "Wafer map defect detection and recognition using joint local and nonlocal linear discriminant analysis," *IEEE Transactions on Semiconductor Manufacturing*, vol. 29, pp. 33–43, Nov 2015.

[59] MIR lab, "WM-811K wafer map dataset." https://www.kaggle.com/qingyi/wm811k-wafer-map, 2018. [Online; accessed April 25, 2019].

[60] M. Piao, C. H. Jin, J. Y. Lee, and J.-Y. Byun, "Decision tree ensemble-based wafer map failure pattern recognition based on radon transform-based features," *IEEE Transactions on Semiconductor Manufacturing*, vol. 31, pp. 250–257, May 2018.

[61] G. Tello, O. Y. Al-Jarrah, P. D. Yoo, Y. Al-Hammadi, S. Muhaidat, and U. Lee, "Deep-structured machine learning model for the recognition of mixed-defect patterns in semiconductor fabrication processes," *IEEE Transactions on Semiconductor Manufacturing*, vol. 31, pp. 315–322, Apr 2018.

[62] T. Nakazawa and D. V. Kulkarni, "Wafer map defect pattern classification and image retrieval using convolutional neural network," *IEEE Transactions on Semiconductor Manufacturing*, vol. 31, pp. 309–314, Jan 2018.

[63] K. Kyeong and H. Kim, "Classification of mixed-type defect patterns in wafer bin maps using convolutional neural networks," *IEEE Transactions on Semiconductor Manufacturing*, vol. 31, pp. 395–402, Aug 2018.

[64] K. Taha, K. Salah, and P. D. Yoo, "Clustering the dominant defective patterns in semiconductor wafer maps," *IEEE Transactions on Semiconductor Manufacturing*, vol. 31, pp. 156–165, Feb 2018.

[65] M. B. Alawieh, F. Wang, and X. Li, "Identifying wafer-level systematic failure patterns via unsupervised learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, pp. 832–844, April 2018.

[66] G. M. Marakas, *Decision Support Systems in the Twenty-first Century*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1999.

[67] D. J. Power, *Decision Support Systems: Concepts and Resources for Managers*. Westport, Conneticut: Quorum Books, 2002.

[68] IBM, "Ibm watson health." https://www.ibm.com/watson/health/. [Online; accessed April 19, 2019].

[69] P. G. W. Keen, *Decision support systems : a research perspective*. Cambridge, Massachusetts: Massachusetts Inistitute of Technology, March 1980.

[70] D. Borenstein, "Towards a practical method to validate decision support systems," *Decision Support Systems*, vol. 23, pp. 227 – 239, 1998.

[71] G. Parmigiani and L. Y. Inoue, *Decision Theory.* Chichester, West Sussex, United Kingdom: John Wiley and Sons, 2009.

[72] A. Kolmogorov, *Grundbegriffe der Wahrscheinlichkeitsrechnung.* Berlin, Heidelberg: Springer, May 1933.

[73] L. A. Shalabi, Z. Shaaban, and B. Kasasbeh, "Data mining: A preprocessing engine," *Journal of Computer Science*, vol. 2, pp. 735–739, Sep 2005.

[74] D. Hawkins, *Identification of outliers.* Chapman and Hall, 1980.

[75] R. Little and D. Rubin, *Statistical Analysis With Missing Data.* Wiley Series in Probability and Statistics - Applied Probability and Statistics Section Series, Wiley, 1987.

[76] S. van Buuren and K. Groothuis-Oudshoorn, "mice: Multivariate imputation by chained equations in r," *Journal of Statistical Software, Articles*, vol. 45, pp. 1–67, 2011.

[77] M. Pleschberger, "Runtime optimization for automated pattern analysis," Master's thesis, Alpen-Adria-Universität Klagenfurt, June 2018.

[78] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, pp. 721–741, Nov 1984.

[79] J. M. Hammersley and P. Clifford, "Markov fields on finite graphs and lattices." [Online; accessed August 25, 2019], 1971.

[80] A. Stuart and K. Ord, *Kendall's Advanced Theory of Statistics, Volume 1: Distribution Theory.* London: Wiley, 1994.

[81] S. Z. Li, *Markov Random Field Modeling in Image Analysis.* London, UK: Springer, 2001.

[82] G. H. Golub and C. F. V. Loan, *Matrix computations.* USA: Johns Hopkins University Press, 1996.

[83] S. R. Garcia and R. A. Horn, *A Second Course in Linear Algebra.* Cambridge Mathematical Textbooks, Cambridge, UK: Cambridge University Press, 2017.

[84] S. Krig, *Interest Point Detector and Feature Descriptor Survey*, pp. 187–246. Cham: Springer International Publishing, 2016.

[85] D. T. Nguyen, W. Li, and P. O. Ogunbona, "Human detection from images and videos: A survey," *Pattern Recognition*, vol. 51, pp. 148 – 175, 2016.

[86] T. Ojala, M. Pietikäinen, and D. Harwood, "Performance evaluation of texture measures with classification based on kullback discrimination of distributions," in *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, vol. 1, pp. 582 – 585 vol.1, 11 1994.

[87] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pp. 886–893, June 2005.

[88] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, pp. 62–66, Jan 1979.

[89] N. M. Zaitoun and M. J. Aqel, "Survey on image segmentation techniques," *Procedia Computer Science*, vol. 65, pp. 797 – 806, 2015. International Conference on Communications, management, and Information technology (ICCMIT'2015).

[90] P. V. C. Hough, "Machine analysis of bubble chamber pictures," *Conf. Proc.*, vol. C590914, pp. 554–558, 1959.

[91] L. Cieplinski, "Mpeg-7 color descriptors and their applications," in *Computer Analysis of Images and Patterns* (W. Skarbek, ed.), (Berlin, Heidelberg), pp. 11–20, Springer Berlin Heidelberg, 2001.

[92] C. Harris and M. Stephens, "A combined corner and edge detector," in *In Proc. of Fourth Alvey Vision Conference*, pp. 147–151, 1988.

[93] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 1150–1157 vol.2, Sep. 1999.

[94] D.-C. He and L. Wang, "Texture unit, texture spectrum and texture analysis," in *12th Canadian Symposium on Remote Sensing Geoscience and Remote Sensing Symposium*, vol. 5, pp. 2769–2772, July 1989.

[95] O. Barkan, J. Weill, L. Wolf, and H. Aronowitz, "Fast high dimensional vector multiplication face recognition," in *IEEE International Conference on Computer Vision*, pp. 1960–1967, Dec 2013.

[96] R. Mehta and K. O. Egiazarian, "Rotated local binary pattern (rlbp) - rotation invariant texture descriptor," in *ICPRAM*, 2013.

[97] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, pp. 129–137, March 1982.

[98] K. Pearson, "On lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, pp. 559–572, 1901.

[99] W. T. Freeman and M. Roth, "Orientation histograms for hand gesture recognition," tech. rep., MERL - Mitsubishi Electric Research Laboratories, Cambridge, MA 02139, Dec. 1994.

[100] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York, US: John Wiley and Sons, 1973.

[101] Chervinskii, "Autoencoder structure." https://upload.wikimedia.org/wikipedia/commons/2/28/Autoencoder_structure.png, Dec 2015. [Online; accessed June 26, 2019].

[102] C.-Y. Liou, J.-C. Huang, and W.-C. Yang, "Modeling word perception using the elman network," *Neurocomputing*, vol. 71, pp. 3150 – 3157, 2008. Advances in Neural Information Processing (ICONIP 2006) / Brazilian Symposium on Neural Networks (SBRN 2006).

[103] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[104] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proceedings of the 31st International Con-*

ference on Machine Learning (E. P. Xing and T. Jebara, eds.), vol. 32 of *Proceedings of Machine Learning Research*, (Bejing, China), pp. 1278–1286, PMLR, 22–24 Jun 2014.

[105] T. Teixeira dos Santos and R. Kern, "Understanding wafer patterns in semiconductor production with variational auto-encoders.," in *Proc. 26th European Symp. on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, Apr 2018.

[106] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[107] S. Kullback and R. A.Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, pp. 79–86, 03 1951.

[108] D. M. Endres and J. E. Schindelin, "A new metric for probability distributions," *IEEE Transactions on Information Theory*, vol. 49, pp. 1858–1860, July 2003.

[109] B. C. Geiger, "A short note on the jensen-shannon divergence between simple mixture distributions," Dec 2018. arXiv:1812.02059[cs.IT].

[110] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, pp. 100–108, 1979.

[111] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.

[112] P. Karsmakers, K. Pelckmans, and J. A. K. Suykens, "Multi-class kernel logistic regression: a fixed-size implementation," in *International Joint Conference on Neural Networks*, pp. 1756–1761, Aug 2007.

[113] F. Santosa and W. Symes, "Linear inversion of band-limited reflection seismograms," *SIAM Journal on Scientific and Statistical Computing*, vol. 7, pp. 1307–1330, 1986.

[114] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, pp. 301–320, 2005.

[115] Q. Zhou, W. Chen, S. Song, J. Gardner, K. Weinberger, and Y. Chen, "A reduction of the elastic net to support vector machines with an application to gpu computing," in *AAAI Conference on Artificial Intelligence*, 2015.

[116] J. Fürnkranz, "Round robin classification," *J. Mach. Learn. Res.*, vol. 2, pp. 721–747, Mar. 2002.

[117] M. ali Bagheri, G. A. Montazer, and S. Escalera, "Error correcting output codes for multiclass classification: Application to two image vision problems," *The 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP 2012)*, pp. 508–513, 2012.

[118] M. Zhang and Z. Zhou, "A review on multi-label learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, pp. 1819–1837, Aug 2014.

[119] W. J. Scheirer, A. Rocha, A. Sapkota, and T. E. Boult, "Towards open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, vol. 35, July 2013.

[120] W. J. Scheirer, L. P. Jain, and T. E. Boult, "Probability models for open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, vol. 36, November 2014.

[121] E. Rudd, L. P. Jain, W. J. Scheirer, and T. Boult, "The extreme value machine," *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, vol. 40, March 2018.

[122] B. B. Dalvi, *Constrained Semi-supervised Learning in the Presence of Unanticipated Classes*. PhD thesis, Carnegie Mellon University, 2015.

[123] B. Dalvi, W. W. Cohen, and J. Callan, "Exploratory learning," in *Machine Learning and Knowledge Discovery in Databases* (H. Blockeel, K. Kersting, S. Nijssen, and F. Železný, eds.), (Berlin, Heidelberg), pp. 128–143, Springer Berlin Heidelberg, 2013.

[124] B. Liu, W. S. Lee, P. S. Yu, and X. Li, "Partially supervised classification of text documents," in *ICML*, vol. 2, pp. 387–394, Citeseer, 2002.

[125] N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall, "Computing gaussian mixture models with em using equivalence constraints," in *NIPS*, 2003.

[126] M. Śmieja and B. C. Geiger, "Semi-supervised cross-entropy clustering with information bottleneck constraint," *Information Sciences*, vol. 421, pp. 254 – 271, 2017.

[127] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, pp. 461–464, 03 1978.

[128] A. Jenul, "Intensity quantification of process patterns in wafer test data," Master's thesis, Alpen-Adria-Universität Klagenfurt, Apr 2019.

[129] P. A. P. Moran, "Notes on continuous stochastic phenomena," *Biometrika*, vol. 37, pp. 17–23, June 1950.

[130] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, pp. 379–423, July 1948.

[131] B. Wang, X. Wang, and Z. Chen, "Spatial entropy based mutual information in hyperspectral band selection for supervised classification," *International Journal of Numerical Analysis and Modeling*, vol. 9, pp. 181–192, Jan 2012.

[132] R Core Team, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2018.

[133] M. Pleschberger, M. Scheiber, and S. Schrunner, "Simulated Analog Wafer Test Data for Pattern Recognition," Jan. 2019.

[134] M. Zhang and Z. Zhou, "A review on multi-label learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, pp. 1819–1837, Aug 2014.

[135] W. Chang, J. Cheng, J. Allaire, Y. Xie, and J. McPherson, *shiny: Web Application Framework for R*, 2019. R package version 1.3.2.

[136] M. Hé, "From NASA to EU: the evolution of the TRL scale in Public Sector Innovation," *The Innovation Journal*, vol. 22, 2017.