

Technologien für die intelligente Automation
Technologies for Intelligent Automation

Jürgen Beyerer
Christian Kühnert
Oliver Niggemann *Editors*

Machine Learning for Cyber Physical Systems

Selected papers from the International
Conference ML4CPS 2018

OPEN



Springer Vieweg

Technologien für die intelligente Automation

Technologies for Intelligent Automation

Band 9

Reihe herausgegeben von

inIT - Institut für industrielle Informa
Lemgo, Deutschland

Ziel der Buchreihe ist die Publikation neuer Ansätze in der Automation auf wissenschaftlichem Niveau, Themen, die heute und in Zukunft entscheidend sind, für die deutsche und internationale Industrie und Forschung. Initiativen wie Industrie 4.0, Industrial Internet oder Cyber-physical Systems machen dies deutlich. Die Anwendbarkeit und der industrielle Nutzen als durchgehendes Leitmotiv der Veröffentlichungen stehen dabei im Vordergrund. Durch diese Verankerung in der Praxis wird sowohl die Verständlichkeit als auch die Relevanz der Beiträge für die Industrie und für die angewandte Forschung gesichert. Diese Buchreihe möchte Lesern eine Orientierung für die neuen Technologien und deren Anwendungen geben und so zur erfolgreichen Umsetzung der Initiativen beitragen.

Weitere Bände in der Reihe <http://www.springer.com/series/13886>

Jürgen Beyerer · Christian Kühnert
Oliver Niggemann
Editors

Machine Learning for Cyber Physical Systems

Selected papers from the International
Conference ML4CPS 2018

OPEN

 Springer Vieweg

Editors

Jürgen Beyerer

Institut für Optronik, Systemtechnik und

Bildauswertung

Fraunhofer

Karlsruhe, Germany

Christian Kühnert

MRD

Fraunhofer Institute for Optronics,

System Technologies and Image Exploitation

IOSB

Karlsruhe, Germany

Oliver Niggemann

inIT - Institut für industrielle

Informationstechnik

Hochschule Ostwestfalen-Lippe

Lemgo, Germany



ISSN 2522-8579

ISSN 2522-8587 (electronic)

Technologien für die intelligente Automation

ISBN 978-3-662-58484-2

ISBN 978-3-662-58485-9 (eBook)

<https://doi.org/10.1007/978-3-662-58485-9>

Library of Congress Control Number: 2018965223

Springer Vieweg

© The Editor(s) (if applicable) and The Author(s) 2019. This book is an open access publication.

Open Access This book is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this book are included in the book's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the book's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer Vieweg imprint is published by the registered company Springer-Verlag GmbH, DE part of Springer Nature
The registered company address is: Heidelberger Platz 3, 14197 Berlin, Germany

Preface

Cyber Physical Systems are characterized by their ability to adapt and to learn. They analyze their environment, learn patterns, and they are able to generate predictions. Typical applications are condition monitoring, predictive maintenance, image processing and diagnosis. Machine Learning is the key technology for these developments.

The fourth conference on Machine Learning for Cyber-Physical-Systems and Industry 4.0 - ML4CPS - was held at the Fraunhofer IOSB in Karlsruhe, on October 23.rd and 24.th 2018. The aim of the conference is to provide a forum to present new approaches, discuss experiences and to develop visions in the area of data analysis for cyber-physical systems. This book provides the proceedings of selected contributions presented at the ML4CPS 2018.

The editors would like to thank all contributors that led to a pleasant and rewarding conference. Additionally, the editors would like to thank all reviewers for sharing their time and expertise with the authors. It is hoped that these proceedings will form a valuable addition to the scientific and developmental knowledge in the research fields of machine learning, information fusion, system technologies and industry 4.0.

Prof. Dr.-Ing. Jürgen Beyerer

Dr.-Ing. Christian Kühnert

Prof. Dr.-Ing. Oliver Niggemann

Table of Contents

	Page
Machine Learning for Enhanced Waste Quantity Reduction: Insights from the MONSOON Industry 4.0 Project <i>Christian Beecks, Shreekantha Devasya, Ruben Schlutter</i>	1
Deduction of time-dependent machine tool characteristics by fuzzy-clustering <i>Uwe Frieß, Martin Kolouch, Matthias Putz</i>	7
Unsupervised Anomaly Detection in Production Lines <i>Alexander Graß, Christian Beecks, Jose Angel Carvajal Soto</i>	18
A Random Forest Based Classifier for Error Prediction of Highly Individualized Products <i>Gerd Gröner</i>	26
Web-based Machine Learning Platform for Condition-Monitoring <i>Thomas Bernard, Christian Kühnert, Enrique Campbell</i>	36
Selection and Application of Machine Learning-Algorithms in Production Quality <i>Jonathan Krauß, Maik Frye, Gustavo Teodoro Döhler Beck, Robert H. Schmitt</i>	46
Which deep artificial neural network architecture to use for anomaly detection in Mobile Robots kinematic data <i>Oliver Rettig, Silvan Müller, Marcus Strand, Darko Katic</i>	58
GPU GEMM-Kernel Autotuning for scalable machine learners <i>Johannes Sailer, Christian Frey, Christian Kühnert</i>	66
Process Control in a Press Hardening Production Line with Numerous Process Variables and Quality Criteria <i>Anke Stoll, Norbert Pierschel, Ken Wenzel, Tino Langer</i>	77
A Process Model for Enhancing Digital Assistance in Knowledge-Based Maintenance <i>Klaudia Kovacs, Fazel Ansari, Claudio Geisert, Eckart Uhlmann, Robert Glawar, Wilfried Sihm</i>	87
Detection of Directed Connectivities in Dynamic Systems for Different Excitation Signals using Spectral Granger Causality <i>Christian Kühnert, Christian Frey, Ruben Seyboldt</i>	97
Enabling Self-Diagnosis of Automation Devices through Industrial Analytics <i>Carlos Paiz Gatica, Alexander Boschmann</i>	107

Making Industrial Analytics work for Factory Automation Applications .	116
<i>Markus Koester</i>	
Application of Reinforcement Learning in Production Planning and Control of Cyber Physical Production Systems	123
<i>Andreas Kuhnle, Gisela Lanza</i>	
LoRaWan for Smarter Management of Water Network: From metering to data analysis	133
<i>Jorge Francés-Chust, Joaquín Izquierdo, Idel Montalvo</i>	



Machine Learning for Enhanced Waste Quantity Reduction: Insights from the MONSOON Industry 4.0 Project

Christian Beecks^{1,2}, Shreekantha Devasya², and Ruben Schlutter³

¹ University of Münster, Germany

christian.beecks@uni-muenster.de

² Fraunhofer Institute for Applied Information Technology FIT, Germany

{christian.beecks,shreekantha.devasya}@fit.fraunhofer.de

³ Kunststoff-Institut Lüdenscheid, Germany

schlutter@kunststoff-institut.de

Abstract. The proliferation of cyber-physical systems and the advancement of Internet of Things technologies have led to an explosive digitization of the industrial sector. Driven by the high-tech strategy of the federal government in Germany, many manufacturers across all industry segments are accelerating the adoption of cyber-physical system and Internet of Things technologies to manage and ultimately improve their industrial production processes. In this work, we are focusing on the EU funded project MONSOON, which is a concrete example where production processes from different industrial sectors are to be optimized via data-driven methodology. We show how the particular problem of waste quantity reduction can be enhanced by means of machine learning. The results presented in this paper are useful for researchers and practitioners in the field of machine learning for cyber-physical systems in data-intensive Industry 4.0 domains.

Keywords: Machine Learning · Prediction Models · Cyber-physical Systems · Internet of Things · Industry 4.0

1 Introduction

The proliferation of cyber-physical systems and the advancement of Internet of Things technologies have led to an explosive digitization of the industrial sector. Driven by the high-tech strategy of the federal government in Germany, many manufacturers across all industry segments are accelerating the adoption of cyber-physical system and Internet of Things technologies to manage and ultimately improve their industrial production processes.

The EU funded project MONSOON⁴ – *M*odel-based *c*o*N*trol *f*ramework for *S*ite-wide *O*ptimizati*ON* of data-intensive processes – is a concrete example where production processes from different industrial sectors, namely process

⁴ <http://www.spire2030.eu/monsoon>

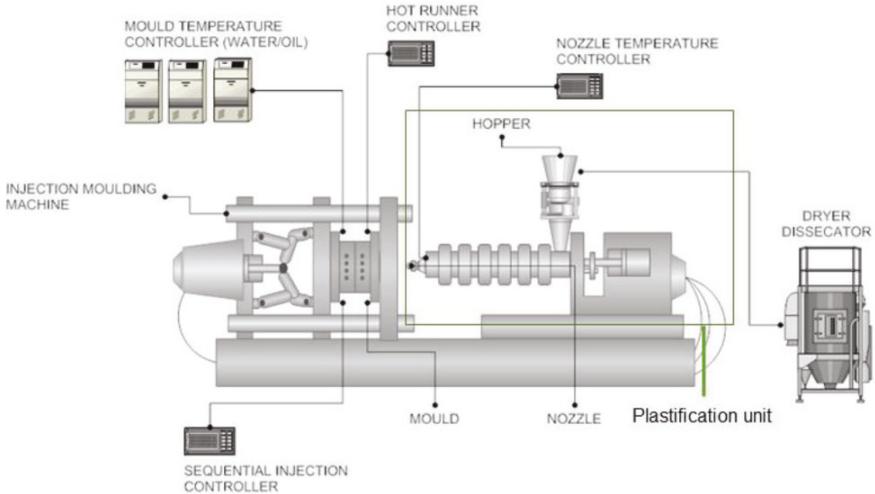


Fig. 1. Parts and periphery of an injection molding machine (KIMW) [2].

industries from the sectors of aluminum and plastic, are to be optimized via data-driven methodology.

In this work, we are focusing on a specific use case from the plastic industry. We use sensor measurements provided by the cyber-physical systems of a real production line producing coffee capsules and aim to reduce the waste quantity, i.e., the number of low-quality production cycles, in a data-driven way. To this end, we model the problem of waste quantity reduction as a two-class classification problem and investigate different fundamental machine learning approaches for detecting and predicting low-quality production cycles. We evaluate the approaches on a data set from a real production line and compare them in terms of classification accuracy.

The paper is structured as follows. In Section 2, we describe the production process and the collected sensor measurements. In Section 3, we present our classification methodology and discuss the results. In Section 4, we conclude this paper with an outlook on future work.

2 Production Process and Sensor Measurements

One particular research focus in the scope of the project MONSOON lies on the plastic sector, where the manufacturing of polymer materials (coffee capsules) is performed by the injection molding method. Injection molding is a manufacturing process that produces plastic parts by injecting raw material into a mold. The process first heats the raw material, then closes the mold and injects the hot plastic. After the holding pressure phase and the cooling phase the mold is opened again and the plastic parts, i.e., coffee capsules in our scenario, are extracted. In this way, each injection molding cycle produces one or multiple

parts. Ideally, the defect rate of each cycle tends toward zero with a minimum waste of raw material. In fact, only cycles with a defect rate below a certain threshold are acceptable to the manufacturer. In order to elucidate the manufacturing process, we schematically show the parts and periphery of a typical injection molding machine in Figure 1. As can be seen in the figure, the injection molding machine comprises different parts, among which the plastification unit builds the core of the machine, and controllers that allow to steer the production process.

The MONSOON Coffee Capsule and Context data set [2] utilized in this work comprises information about 250 production cycles of coffee capsules from a real injection molding machine. It contains 36 real-valued attributes reflecting the machine's internal sensor measurements for each cycle. These measurements include values about the internal states, e.g. temperature and pressure values, as well as timings about the different phases within each cycle. In addition, we also take into account quality information for each cycle, i.e., the number of non-defect coffee capsules which changes throughout individual production cycles. If the number of produced coffee capsules is larger than a predefined threshold, we label the corresponding cycle with *high.quality*, otherwise we assign the label *low.quality*. The decision about the quality labels was made by domain experts.

Based on this data set, we benchmark different fundamental machine learning approaches and their capability of classifying low-quality production cycles based on the aforementioned sensor measurements. The methodology and results are described in the following section.

3 Application of Machine Learning in Plastic Industry

By applying machine learning to the sensor measurements gathered from a production line of coffee capsules equipped with cyber-physical systems, we aim at detecting and predicting low-quality production cycles. For this purpose, we first preprocess the data by centering and scaling the attributes and additionally excluding attributes with near zero-variance. Preprocessing was implemented in the programming language *R* based on the *CARET* package [7].

Based on the preprocessed data set, we measured the classification performance in terms of *balanced accuracy*, *precision*, *recall*, and *F1* via k-fold cross validation, where we set the number of folds to a value of 5 and the number of repetitions to a value of 100. That is, we used 80% of the data set as training data and the remaining 20% as testing data for predicting the quality of the production cycles. We averaged the performance over 100 randomly generated training sets and test sets.

We investigated the following fundamental predictive models, all implemented via the *CARET* package in *R*:

- *k-Nearest Neighbor* [4]: A simple non-parametric and thus model-free classification approach based on the Euclidean distance.
- *Naive Bayes* [5]: A probabilistic approach that assumes the independence of the attributes.

- *Classification and Regression Trees* [9]: A decision tree classifier that hierarchically partitions the data.
- *Random Forests* [3]: A combination of multiple decision trees in order to avoid over-fitting.
- *Support Vector Machines* [11]: An approach that aims to separate the classes by means of a hyperplane. We investigate both linear SVM and SVM with RBF kernel function.

We evaluated the classification performance of the predictive models described above based on the injection molding machine’s internal states which are captured by the sensor measurements. The corresponding classification results are summarized in Table 1.

Table 1. Classification results of different predictive models.

	balanced accuracy	precision	recall	F1
k-NN	0.697	0.638	0.686	0.657
Naive Bayes	0.643	0.604	0.563	0.578
CART	0.637	0.595	0.566	0.573
Random Forest	0.653	0.619	0.570	0.589
SVM (linear)	0.632	0.626	0.488	0.540
SVM (RBF)	0.663	0.643	0.563	0.594

As can be seen from the table above, all predictive models reach a classification accuracy of at least 63%, while the highest classification accuracy of approximately 69% is achieved by the k-Nearest Neighbor classifier. For this classifier, we utilized the Euclidean distance and set the number of nearest neighbors k to a value of 7. In fact, the k-Nearest Neighbor classifier is able to predict the correct quality labels for 172 out of 250 cycles on average.

It is worth noting that this rather low classification accuracy (69%) might have a high impact on the real production process, since in our particular domain hundreds of coffee capsules are produced every minute such that even a small enhancement in waste quantity reduction will lead to a major improvement in production costs reduction. In addition, we have shown that the performance of the k-Nearest Neighbor classifier can be improved to value of 72% when enriching the sensor measurements with additional process parameters [2].

To conclude, the empirical results reported above indicate that even a simple machine learning approach such as the k-Nearest Neighbor classifier is able to predict low-quality production cycles and thus to enhance the waste quantity reduction. Although the provided sensor measurements are of limited extent regarding the number of measurements, we believe that our investigations will be helpful for further data-driven approaches in the scope of the project MONSOON and beyond.

4 Conclusions and Future Work

In this work, we have focused on the EU funded project MONSOON, and have shown how the particular problem of waste quantity reduction can be enhanced by means of machine learning. We have applied fundamental machine learning methods to the sensor measurements from a cyber-physical system of a real production line in the plastic industry and have shown that predictive models are able to exploit optimization potentials by predicting low-quality production cycles. Among the investigated predictive models, we have empirically shown that the k-Nearest Neighbor classifier yields the highest prediction performance in terms of accuracy.

As future work, we aim at investigating different preprocessing methods and ensemble strategies in order to improve the overall classification accuracy. We also intend to evaluated different distance-based similarity models [1] for improving the performance of the k-Nearest Neighbor classifier. In addition, we intend to extend our performance analysis to other industry segments, for instance the production of surface-mount devices [10], and to investigate metric access methods [8, 12] as well as ptolemaic access methods [6] for efficient and scalable data access.

5 Acknowledgements

This project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 723650 - MONSOON. This paper reflects only the authors views and the commission is not responsible for any use that may be made of the information it contains. It is based on a previous paper [2].

References

1. Beecks, C.: Distance based similarity models for content based multimedia retrieval. Ph.D. thesis, RWTH Aachen University (2013)
2. Beecks, C., Devasya, S., Schlutter, R.: Data mining and industrial internet of things: An example for sensor-enabled production process optimization from the plastic industry. In: International Conference on Industrial Internet of Things and Smart Manufacturing (2018)
3. Breiman, L.: Random forests. *Machine learning* **45**(1), 5–32 (2001)
4. Cover, T., Hart, P.: Nearest neighbor pattern classification. *IEEE transactions on information theory* **13**(1), 21–27 (1967)
5. Domingos, P., Pazzani, M.: On the optimality of the simple bayesian classifier under zero-one loss. *Machine learning* **29**(2), 103–130 (1997)
6. Hetland, M.L., Skopal, T., Lokoč, J., Beecks, C.: Ptolemaic access methods: Challenging the reign of the metric space model. *Information Systems* **38**(7), 989–1006 (2013)
7. Kuhn, M.: Building predictive models in r using the caret package. *Journal of Statistical Software, Articles* **28**(5), 1–26 (2008)

8. Samet, H.: Foundations of multidimensional and metric data structures. Morgan Kaufmann (2006)
9. Steinberg, D., Colla, P.: Cart: classification and regression trees. The top ten algorithms in data mining **9**, 179 (2009)
10. Tavakolizadeh, F., Soto, J., Gyulai, D., Beecks, C.: Industry 4.0: Mining physical defects in production of surface-mount devices. In: Industrial Conference on Data Mining (2017)
11. Vapnik, V.: The nature of statistical learning theory. Springer science & business media (2013)
12. Zezula, P., Amato, G., Dohnal, V., Batko, M.: Similarity search: the metric space approach, vol. 32. Springer Science & Business Media (2006)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Deduction of time-dependent machine tool characteristics by fuzzy-clustering

Uwe Friess^{1*}, Martin Kolouch¹ and Matthias Putz

¹ Fraunhofer Institute for Machine Tools and Forming Technology IWU, Chemnitz, Germany

* Corresponding author. Tel.: +49-371-5397-1393; fax: +49-371-5397-6-1393;
E-mail address: uwe.friess@iwu.fraunhofer.de

Abstract. With the onset of ICT and big data capabilities, the physical asset and data computation is integrated in manufacturing through Cyber Physical Systems (CPS). This strategy also denoted as Industry 4.0 will improve any kind of monitoring for maintenance and production planning purposes. So-called big-data approaches try to use the extensive amounts of diffuse and distributed data in production systems for monitoring based on artificial neural networks (ANN). These machine learning approaches are robust and accurate if the data base for a given process is sufficient and the scope of the target functions is curtailed. However, a considerable proportion of high-performance manufacturing is characterized by permanently changing process, workpiece and machine configuration conditions, e.g. machining of large workpieces is often performed in batch sizes of one or of a few parts. Therefore, it is not possible to implement a robust condition monitoring based on ANN without structured data-analyses considering different machine states – e.g. a certain machining operation for a certain machine configuration. Fuzzy-clustering of machine states over time creates a stable pool representing different typical machine configuration clusters. The time-depending adjustment and automatized creation of clusters enables monitoring and interpretation of machine tool characteristics independently of single machine states and pre-defined processes.

Keywords: Fuzzy logic, Machine tool, Machine learning, Clustering.

1 Introduction

Technological value adding by extracting of CPS-capabilities is acting as selective pressure not only at academic levels but already on the shop floor [1-3]. Integrally modules are predictive maintenance and cloud-based monitoring of production systems [4-6]. In [7] and [8] the authors introduced an approach to overcome limits in condition monitoring of large and special-purpose machine tools. The core challenge to address is the time-based change in nearly every internal and external constraint parameter (**Fig. 1**).

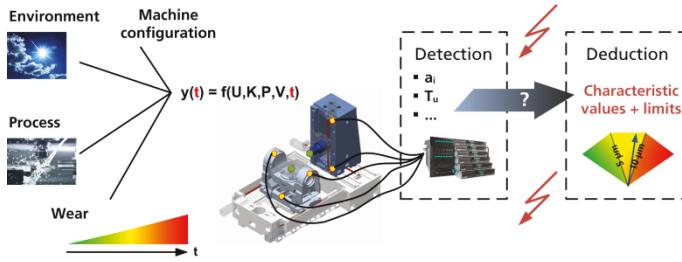


Fig. 1. Challenges in deduction of limits based on measuring data

This results in difficulties to correlate any kind of measuring data with the health state of the machine and its components. Measures to address these challenges are:

1. Definition of Machine States (MSs) based on trigger parameters (TPs) (**Table 1**).
2. Deduction and comparison of Characteristic Values (CVs) is only carried out
 - a. for the same machine state
 - b. Gradually for a cluster resulting from the fuzzy-clustering (see 5 below)
3. Deduction of dynamic limits for the CVs over time
4. Fuzzy-based interpretation of the current CV-values regarding their expectation values (see section 5, **Fig. 5**)
5. Fuzzy-Clustering of MSs to create a stable pool including a broad range of characteristic configurations of the machine tool

1.1 Limits of cluster analyses based on pre-defined machine states

The fuzzy clustering of pre-defined MSs can be adequate for monitoring of components with clear objectives, e.g. the health state. Essential basis is a balanced definition of MSs by a maintenance expert. Therefore the pre-definition of MSs is prone to an unexperienced workforce. More challenging is the altering of processes and work-piece batches which leads to a decay of the initial defined MSs. The expert therefore needs to define new relevant MSs and exclude old ones from the “pool” (see Fig. 9 in [8]).

Further potentials can be obtained if the pre-definition of MSs is replaced by an auto-derivation of MSs and a subsequent fuzzy clustering of these MSs with the objective of a broad characterization of the machine tool configurations over time. For this purpose, a tree-step machine-learning cycle is introduced subsequently and described in the following sections:

1. Auto-definition of MS by segmentation of MS parameters (section 2)
2. Deriving of Characteristic Values (CVs) for every state as described in [8]
3. MS-TP-reduction: Correlation analyses between MSs, CVs, parameter reduction and exclusion of non-significant MSs (section 3 and 4)
4. Fuzzy-clustering of MSs including derivation of Cluster-CVs (section 5)
5. Deriving of machine-characterizing Clusters which represent concrete categories of machine tools, e.g. heavy machining for certain feed axes configuration.

2 Auto-definition of MSs by segmentation of TPs for different parameter numbers

A typical pre-defined MS is characterized by a subset of TPs as presented in [7] (**Table 1**). The MSs depict in Table 1 are represented by using different TPs for an axis stroke (see **Fig. 2**).

Table 1. Normalized data of MSs using the relative normalization of TP, overall cycle.

MS	1	2	3	4	5	6	7	8	9
TP									
1.1 Automatic mode	1	1	1	1	1	1	1	1	1
3.1 x-pos.	1	1	1	1	1	1	1	1	1
4.1 y-pos.	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
4.2 y-pos. (Δ)	0	0	0	0	0	0	0	0	0
5.1 z-pos.	1	0.5	0	0.86	0.41	0.14	0.05	0.55	0.95
6.1 Jerk	1	1	1	1	1	1	1	1	1
7.1 Acceleration	1	1	1	0.5	0.5	0	0.75	0.75	0.75
8.1 Feed rapid traverse	1	1	1	0	0.67	0.83	0.67	1	1
9.1 Temperature of y2 ball-screw nut	0	0.40	0.66	0.81	0.96	0.91	1	0.71	0.70

TPs can vary in a broad range, e.g. the current position of an axis or the feed. A combination that doesn't occur in praxis – e.g. a stroke between 0 and 1 mm for a given axis – is not detectable and therefore it does not increase the complexity. However an axis stroke of 1000 mm could be divided from any numerical integer between 2 and ∞ in principle. Thus it is still necessary to have an upfront definition of TPs ranges. A practical solution for dynamic TPs like the jerk, the acceleration or the feed consists in definition of altering-constraints to intersect a MS in sub-phases.

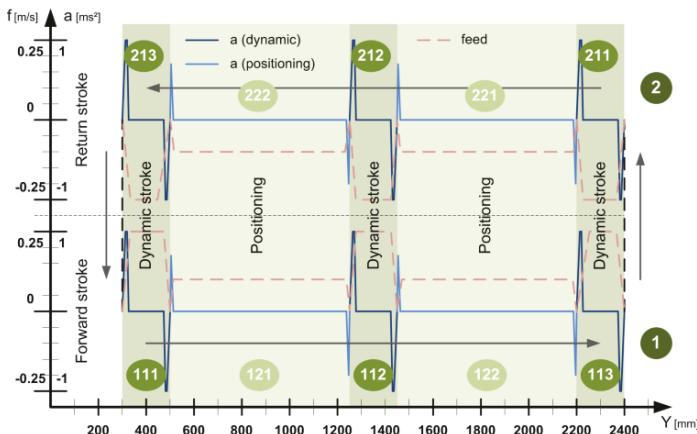
A MS is not a singular event but a process which is characterized by a given timespan. Real-life processes of machine tools are continuous and can be fragmented in several sub-phases by various measures. An example would be a boring operation with a specific tool. Another one could be the stroke of a single axis as depicted in **Table 2** and **Fig. 2**.

The definition of an overall process is complex and may vary depending on the desired application or monitoring object. This process would be the highest level of a MS as depict in **Table 1**. The y-axis executes a stroke from 300 mm up to 2400 mm and back, therefor representing a complete cycle. This overall stroke can consequently be divided into several sup-phases which can be treated as discrete MS. These “sub-MS” can be identified in dependence of the altering of dynamic parameters as described in **Table 2**. To distinguish them from each other every sub-MS is described by numerical values depending on the level of the dynamic parameter (**Table 2**, left). Alternative identifications are also conceivable. However the introduced description based on levels links physical parameters directly to the sub-MSs.

Table 2. Levels of MSs in dependence of the dynamic y-axis stroke.

Level	Description (numbers in [mm])						Length [mm]	Number of MS per level
0	1	2	3	4	5	6		
0	0	0	0	0	0	0	Overall stroke	2x2100
1	0	0	0	0	0	0	Forward stroke (FS)	2100
2	0	0	0	0	0	0	Backward stroke (BS)	2100
1	1	1	0	0	0	0	FS, dynamic phase (DP), 300-500	200
1	1	2	0	0	0	0	FS, DP, 1250-1450	200
1	1	3	0	0	0	0	FS, DP, 2200-2400	200
1	2	1	0	0	0	0	FS, positioning (PO), 500-1250	750
1	2	2	0	0	0	0	FS, PO, 1450-2200	750
2	1	1	0	0	0	0	BS, DP, 2400-2200	200
...
2	2	2	0	0	0	0	BS, PO, 1250-500	750
1	1	1	1	1	0	0	FS, DP, acceleration (AC), 300-(~)375	75
1	1	2	1	2	0	0	FS, DP, AC, 1250-(~)1325	75
...	30
1	1	1	2	1	0	0	FS, DP, constant feed (CF), (~)375-(~)425	170
...
1	1	1	1	1	1	1	FS, DP, AC, positive jerk (PJ), 300-(~)304	3,33 (theor.)
...	50

If the lowest possible level is defined by the direction of the jerk, a maximum of 50 sub-phases can be identified based on path dynamics. We divide the overall stroke in 12 sub-phases based on the identification levels 1-3 of **Table 2** for demonstration purposes as depicted in **Fig. 2**. Practically other TPs like the dynamic path of a second axis as well as process parameters could also vary in parallel.

**Fig. 2.** Test cycle used in [8] including sub-phases of MSs

Obviously the auto-detection of any possible MS based on time-dependent changes of any considered TP is not a practicable solution. Therefore a parallelization approach is suggested, where MPs based on different TPs for different sub-phases – down until the level where the TPs still vary – are created, CVs derived and correlation analyses between MSs and TPs carried out. This overall approach is depicted in **Fig. 3**.

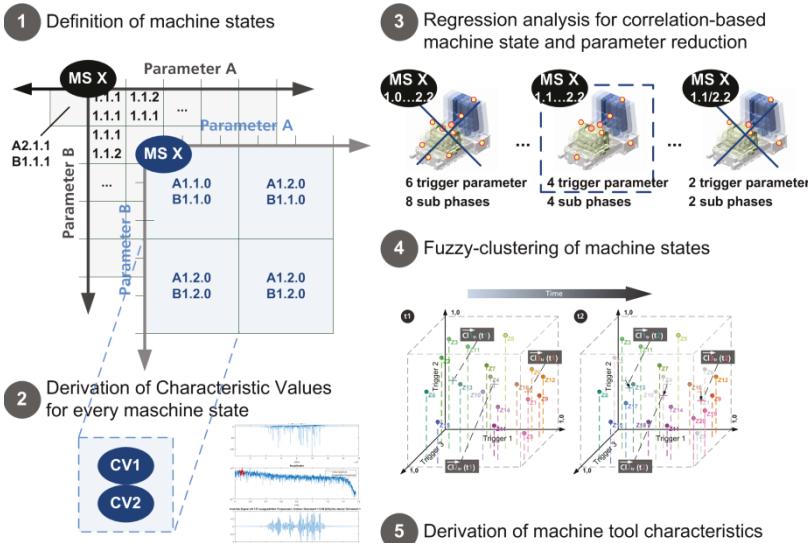


Fig. 3. Suggested approach for automatic MS- and TP reduction

3 Regression analysis for correlation-based machine state and parameter reduction

The fuzzy clustering of MSs, as presented in [8] can be exercised without any consideration of possible correlations between TPs and CVs. This is possible for a limited number of pre-defined MSs based on practical considerations about components of interest and – heuristically anticipated – correlations between CVs and TPs. If a broad range of TPs is combined with a variable resolution of TP sections as well as time spans the clustering of all combinations – for every CV – becomes unpractical, statistically challenging and the information content decays. Therefore a reduction of significant MS and TPs for these states is necessary. This task can be addressed by the usage of an artificial neural network (ANN), but the robustness and accuracy of such depends heavily on the quantity of training data. This means that every relevant MS has to occur several times before the ANN can play off its strength. This is not a given in non-serial machine tool applications as described in section 1.

For this purpose, regression analysis between the TPs and the CVs can be employed as suggested in this paper. Based on the introduced cycle, a regression analysis was carried out. The input variables (TPs) and the responses (CVs) used in the regres-

sion analysis are shown in Table 8. This includes all varying parameters of the MS. The considered MS regression analysis does not aim to a quantification of the regression function between the input variables and the responses but it should statistical validate the significance of the input variables (for more detail see [9]). Thus, a linear function without any interactions is chosen for the regression analysis.

Table 3. Defined input variables and responses in the regression analysis

Input variables = TPs	Responses = CVs
z-position	Effective vibration level
Acceleration	Frequency of the highest peak
Feed rapid traverse	
Temperature of the ball-screw nut	

The included MSs are 10 sub-phases of **Fig. 2** for every TP-combination of **Table 1**. Sup-phases 113 and 213 (**Fig. 2**) are not considered due to their corrupted measurement data. It should be noted the TPs 4.1 and 4.2 vary in accordance to the sub-phases. Therefore 90 different – but related – MS are taken into account.

4 Practical example

The test cycle of **Fig. 2** was derived for the 9 MS in **Table 1** (**Fig. 4**). 51 cycles were successively executed for each MS, resulting in an overall time of 2550s. Every cycle includes all sub-phase (“sub-MS”) of **Fig. 2**.



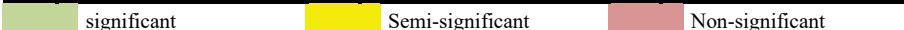
Fig. 4. UNION PCR130 machine; y- and z-axis used for the test cycles

Based on these cycles, a linear regression analyses was derived for the sub-phases using the commercial software Cornerstone®. The aim of the regression analyses is not to derive a quantitative model with the aim to predict the CVs based on the TPs. The data available is not sufficient for such a purpose. The regression model is only linear and not representative for the TPs as well as the CVs overall range. However, the regression analysis deducts significance terms for every input-parameter (= TP), therefore distinguishing the relevant TPs for a given CV (responses in **Table 4**) from the irrelevant ones. Furthermore, when comparing the significance terms of the TPs with the adjusted R-Square value of the correlation analysis we obtain an assessment

to define adequate sub-phases. Additionally the correlation between the significant TPs (Covariance matrix) is checked to exclude TPs with high covariance's. For example the temperature has an even higher significance-term in sub-phase 112 than the feed. However the Covariance matrix indicates that the temperature is highly correlated to the Temperature (-0.9861) and should therefore be excluded for the subsequent clustering for the CV fmax. Successively the number of relevant MSs is significantly reduced. The number of relevant TPs is simultaneously reduced. **Table 4** depicts the overall result for all 10 sub-phases and 4 inputs, carried out separately for each of the 9 MSs from **Table 1**.

Table 4. Correlation analysis results for the sub phases of MS 1-9 and both CVs.

Correlation analysis			Significance Terms (of inputs/TPs)					Quality of Regression			
Levels of sub phases			Responses (= CVs)	Constant	Z-position	acceleration a	Feed f	Temp.	R-Square	Adjusted R-Square	RMS Error
1st	2nd	3rd									
0	0	Peff	0.010	0.494	0.043	0.054	0.011	0.790	0.664	0,067	
		fmax	0.277	0.008	0.691	0.095	0.379	0.818	0.757	4,541	
	1	Peff	2e-05	0.977	0.635	0.005	1e-05	0.966	0.954	0.028	
		fmax	3e-09	0.769	0.535	0.135	0.379	0	0	8.054	
	1	Peff	1e-04	0.756	0.052	0.011	9e-05	0.934	0.913	0.035	
		fmax	3e-05	0.731	0.198	2e-4	2e-05	0.960	0.947	2.366	
	2	Peff	0.014	0.061	0.196	0.323	0.013	0.677	0.569	0.029	
		fmax	6e-05	0.088	0.683	0.407	0.204	0.359	0.267	9.828	
	1	Peff	0.011	0.052	0.158	0.356	0.010	0.698	0.597	0.039	
		fmax	9e-06	0.095	0.181	0.813	0.355	0.347	0.254	7.248	
0	0	Peff	0.059	0.460	0.132	0.051	0.071	0.548	0.398	0.114	
		fmax	0.023	0.047	0.237	0.001	0.032	0.921	0.873	4.527	
	1	Peff	0.001	0.439	0.156	0.013	0.001	0.845	0.793	0.038	
		fmax	0.519	0.991	0.880	0.002	0.517	0.770	0.738	3.076	
	1	Peff	2e-04	0.128	0.588	0.002	1e-04	0.926	0.902	0.054	
		fmax	0.550	0.861	0.802	0.001	0.499	0.806	0.778	2.903	
	2	Peff	3e-05	0.895	0.687	0.485	3e-05	0.931	0.921	0.009	
		fmax	1e-04	0.041	0.619	0.163	0.168	0.471	0.396	9.440	
	2	Peff	0.002	0.004	0.997	0.006	0.901	0.829	0.772	0.015	
		fmax	5e-05	0.047	0.608	0.207	0.286	0.452	0.373	8.336	

 significant Semi-significant Non-significant

Several important conclusions can be detracted from the results of the correlation analysis and the subsequent survey of Covariance matrix of the significant TPs:

- The most promising sub-phases with the best correlations are the dynamic phases in the middle of the axis stroke; the auto-definition detects this sub-phase MSs
- The effective vibration level is clearly correlated to the temperature of the nut
- The ball pass frequency of the ball-screw nut outer ring is clearly correlated to the feed (the frequency can be calculated based on geometric parameters)

- The quality of regression for the effective Vibration level (Peff) is significant in more sub-phases and therefore more generally usable than the ball pass frequency of the ball-screw (Y2)nut (fmax)

Therefore the auto detection mechanism would choose sub-phases 112 and 212 as most relevant for monitoring. In regard to the CVs, the temperature remains the only relevant TP for the effective Vibration level while the feed remains the only relevant TP for the outer ring frequency of the ball-screw nut.

5 Deduction of machine characteristics based on clustering

The clustering was deducted solely on base of the two relevant TPs for each of the two CVs as described in section 4. The algorithm is described in detail in [8] based on [9]. Every MS is gradually attributed to the cluster centres. The relevant TP 8.1 and 9.1 do not vary in accordance to the sub-phases, so the clustering solely depends on the (average) TP of the 9 MS. We obtain cluster centres at 0.71/0.99/0.00 for TP 8.1 (feed rapid traverse) respectively 0.09/0.92/0.64 for TP 9.1 (temperature of y2 ball screw nut). **Table 5** depicts the TP-values for each MS and their affiliation rate.

Table 5. Normalized TP and affiliation rates per cluster for all MS; optimization cycle $n_{opt} = 100$; fuzzifier $w = 1.5$

Maschine states	1	2	3	4	5	6	7	8	9
Relevant TPs									
8.1 Feed rapid traverse (for CV2)	1	1	1	0	0.67	0.83	0.67	1	1
9.1 Temperature of y2 ball-screw nut (for CV1)	0	0.40	0.66	0.81	0.96	0.91	1	0.71	0.70
Cluster	Affiliation rates per cluster								
1	TP 8.1	1	0	0	0	1	0.732	1	0
	TP 9.1	1	0.273	0	0	0	0	0	0
2	TP 8.1	0	1	1	0	0	0.268	0	1
	TP 9.1	0	0.034	0	0.857	1	1	0.997	0.013
3	TP 8.1	0	0	0	1	0	0.000	0	0
	TP 9.1	0	0.693	1	0.143	0	0	0.987	0.994

Based on the affiliation rates of each MS the clusters represent typical CV-progressions as depicted in **Fig. 5** for CV1 (effective vibration level). We obtain several alarms for cluster 1 (**Fig. 5 left**) with limits corresponding to a band in the $\pm 3\sigma$ range. This is due to the fact that cluster 1 represents the head-up of the machine tool representing an unsettled pool of MSs (essentially MS 1). Alternatively a band of $\pm 6\sigma$ for limit calculation can be used.

The auto-reduction of relevant TP and MS generates clusters which represent typical conditions of a machine tool. When combined with CV-information's and by subsequent structure-attribution the gathering of machine tool characteristics over time is achievable.

A possible example includes the CV1 (effective vibration level) which represents “undesired system energy” and causes wear. Therefore the CV1-level should be observed. The number and range of MS will gradually improve over time for a given machine tool. Therefore more and more clusters arise. Some of these clusters represent high wear-proceeding defined by high CV1-levels and caused by higher-than-average bearing temperatures while others won't. Consequently machining operations as well as manufactured parts can be categorized and evaluated regarding their wear-processing characteristics. While some correlations may state the obvious – e.g. heavy machining – the overall load-wear correlation of the machine tool becomes more transparent. Furthermore measurements like switching of an axis position for high wear-processing manufactured parts became practicable.

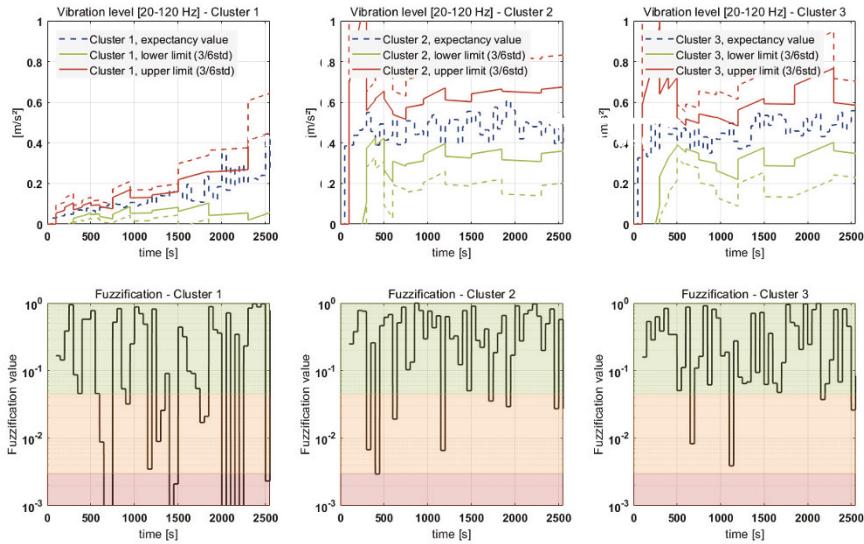


Fig. 5. Cluster-CV progress including Fuzzification ; CV1: Peff of ball-screw nut of Y2 axis

6 Conclusion

The auto-definition of relevant MS is crucial for addressing the ongoing changes in internal and external conditions of large and special purpose machine tools. By using a linear regression a significant reduction on the number of MS is possible. This includes the distinction between relevant and irrelevant sub-phases. Furthermore the regression analysis also enables to reduce the number of relevant input TPs (e.g. measuring parameters) per CV.

Based on a subsequent clustering of the machine states these clusters represent a more stable base than a single MS. Their specific TP-ranges in context of specific CVs (e.g. a ball-pass frequency) represent machine tool characteristics. A categorization of processes and manufactured parts – regarding their wear-processing as well as

quality stability – becomes possible when combined with structural information's and a process-evaluation regarding their cluster attribution.

Further research is necessary due to different clustering approaches as well as more complex regression model approaches (e.g. quadratic). Furthermore, the deduction of complex Characteristic values for entire structural components using several CVs based on different algorithms will be investigated.

Acknowledgements

The research presented in this paper is funded by the European Union (European Social Fund) and by the Free State of Saxony. The authors would like to thank the founders.

References

1. Lee, J.; Bagheri, B.; Kao, H.-A.: "A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems", *Manufacturing Letters*. 18–23 2015.
2. Lu, Y.: Industry 4.0: a survey on technologies, applications and open research issues. *Journal of Industrial Information Integration* 6, 1-10 (2017)
3. Gausemeier, J.; Klocke, F.: Industrie 4.0 – International Benchmark, Options for the Future and Recommendations for Manufacturing Research, Paderborn 2016.
4. J. T. Farinha, I. Fonseca, R. Oliveira und H. Raposo, „CMMS – An integrated view from maintenance management to on-line condition monitoring.“ in *Proceedings of Maintenance Performance Measurement and Management (MPMM) Conference*, Coimbra, Portugal, 2014.
5. R. Teti, K. Jemielniak, G. O'Donnell and D. Dornfeld, „Advanced monitoring of machining operations,“ *CIRP Annals - Manufacturing Technology*, Nr. 59, pp. 717-739, 2010.
6. W. Derigent, E. Thomas, E. Levrat and B. Jung, „Opportunistic maintenance based on fuzzy modelling of component Proximity,“ *CIRP Annals - Manufacturing Technology*, Bd. 58, pp. 29-32, 2009.
7. M. Putz, U. Frieß, M. Wabner, A. Friedrich, A. Zander and H. Schlegel, „State-based and self-adapting Algorithm for Condition Monitoring,“ in *10th CIRP Conference on Intelligent Computation in Manufacturing Engineering - CIRP ICME '16*, Ischia, Naples, Italy, 20 - 22 July 2016
8. U. Frieß, M. Kolouch, M. Putz, A. Friedrich and A. Zander: “Fuzzy-clustering of machine states for condition monitoring”, *CIRP Journal of Manufacturing Science and Technology*, Vol. XX, xxx-xxx, 2018.
9. R. Kruse, C. Borgelt, C. Braune, F. Klawonn, C. Moewes und M. Steinbrecher, *Computational Intelligence - Eine methodische Einführung in Künstliche Neuronale Netze, Evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze*, Wiesbaden: Springer Vieweg, 2. Auflage 2015.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Unsupervised Anomaly Detection in Production Lines

Alexander Graß, Christian Beecks, Jose Angel Carvajal Soto

Fraunhofer Institute for Applied Information Technology FIT, Germany
`{alexander.grass, christian.beecks, angel.carvajal}@fit.fraunhofer.de`

Abstract. With an ongoing digital transformation towards industry 4.0 and the corresponding growth of collected sensor data based on cyber-physical systems, the need for automatic data analysis in industrial production lines has increased drastically. One relevant application scenario is the usage of intelligent approaches to anticipate upcoming failures for maintenance. In this paper, we present a novel approach for anomaly detection regarding predictive maintenance in an industrial data-intensive environment. In particular, we are focusing on historical sensor data from a real reflow oven that is used for soldering surface mount electronic components to printed circuit boards. The sensor data, which is provided within the scope of the EU-Project COMPOSITION (under grant no. 723145), comprises information about the heat and the power consumption of individual fans inside a reflow oven. The data set contains time-annotated sensor measurements in combination with additional process information over a period of more than seven years.

Keywords: Unsupervised Learning, Industry 4.0, Anomaly Detection

1 Introduction

In the last couple of years, the importance of cyber-physical systems in order to optimize industry processes, has led to a significant increase of sensorized production environments. Data collected in this context allows for new intelligent solutions to e.g. support decision processes or to enable predictive maintenance. One problem related to the latter case is the detection of anomalies in the behavior of machines without any kind of predefined ground truth. This fact is further complicated, if a reconfiguration of machine parameters is done on-the-fly, due to varying requirements of multiple items processed by the same production line. As a consequence, a change of adjustable parameters in most cases directly leads to divergent measurements, even though those observations should not be regarded as anomalies.

In the scope of the EU-Project COMPOSITION (under grant no. 723145), the task of detecting anomalies for predictive maintenance within historical sensor data from a real reflow oven was investigated. While the oven is used for soldering surface mount electronic components to printed circuit boards based on continuously changing recipes, one related problem was the unsupervised recognition

of potential misbehaviors of the oven resulting from erroneous components. The utilized data set comprises information about the heat and power consumption of individual fans. Apart from additional machine parameters like a predefined heat value for each section of the oven, it contains time-annotated sensor observations and process information recorded over a period of more than seven years.

As one solution for this problem, in the upcoming chapters we will present our approach named Generic Anomaly Detection for Production Lines, short GADPL. After a short introduction on related approaches, in the upcoming chapters we will focus on a description of the algorithm. Afterwards we outline the evaluation carried out on the previously mentioned project data, followed by a concluding discussion on the approach and future work.

2 Related Work

While the topic of anomaly detection and feature extraction is covered by a broad amount of literature, in the following we will focus on a selection of approaches that led to the here presented algorithm. Recently, the automatic description of time series, in order to understand the behavior of data or to perform subsequent operations has drawn the attention of many researchers. One idea in this regard is the exploitation of Gaussian processes [3, 5] or related structural compositions [4]. Here, a time series is analyzed using a semantically intuitive grammar consisting of a kernel alphabet. Although corresponding evaluations show impressive results, they are rather applicable to smaller or medium sized historical data, since the training of models is comparatively time consuming. In contrast, other approaches exist, which focus on the extraction of well-known statistical features, further optimized by means of an additional feature selection in a prior stage [2]. However, the selection of features is evaluated based on already provided knowledge and thus not applicable in unsupervised use-cases. A last approach discussed here, uses the idea of segmented self-similarity joins based on raw data [7]. In order to decrease the complexity, segments of a time series are compared against each other in the frequency domain. Even though this idea provides an efficient foundation for many consecutive application scenarios, it lacks the semantic expressiveness of descriptive features as it is the case for the already mentioned methods.

In the upcoming chapter we consequently try to deal with those challenges, while presenting our approach for unsupervised anomaly detection.

3 Approach

The hereafter presented description of GADPL is based on the stage-wise implementation of the algorithm. After an initial clustering of similar input parameters (3.1) and a consecutive segmentation (3.2), we will discuss the representation of individual segments (3.3) and the corresponding measurement of dissimilarity

(3.4). GADPL is also summarized in figure Algorithm 1, at the end of this chapter.

3.1 Configuration Clustering

In many companies, as well as in the case of COMPOSITION, a single production line is often used to produce multiple items according to different requirements. Those requirements are in general defined by varying machine configurations consisting of one or more adjustable parameters, which are changed 'on-the-fly' during runtime. For a detection of deviations with respect to some default behavior of a machine, this fact raises the problem of invalid comparisons between sensor measurements of dissimilar configurations. If a measurement or an interval of measurements is identified as an anomaly, it should only be considered as such, if this observation is related to the same configuration as observations representing the default behavior. In other words:

If $C_k = \{x_l := \lambda_l | 1 \leq l \leq M\}$ is a configuration with M parameters x_l of value λ_l , then for the dissimilarity δ of two measurement representations $y_{1,i}$ and $y_{2,j}$ with associated configurations C_i and C_j , it holds that:

$$\delta(y_{1,i}, y_{2,j}) \text{ is defined} \quad \text{iff.} \quad i == j$$

Therefore in advance to all subsequent steps, at first all sensor measurements have to be clustered according to their associated configuration.

For simplicity, in the following subsections we are only discussing the process within a single cluster, although one has to keep in mind, that each step is done for all clusters in parallel.

3.2 Segmentation

As a result of the configuration-based clustering, the data is already segmented coarsely. However, since this approach describes unsupervised anomaly detection, the idea of a further segmentation is, to create some kind of ground truth, which reflects the default behavior of a machine. In subsection 3.4 we will see, how the segmentation is utilized to implement this idea. In an initial step, a maximum segmentation length is defined, in order to specify the time horizon, after which an anomaly can be detected. Assuming a sampling rate of 5mins per sensor, the maximum length of a segment would consequently be $(60 \cdot 24)/5 = 288$ to describe the behavior on a daily basis. Although a decrease of the segment length implies a decrease of response time, it also increases the computational complexity and makes the detection more sensitive to invalid sensor measurements. In this context, it needs to be mentioned that in this stage segments are also spitted, if they are not continuous with respect to time as a result of missing values. Another fact that has to be considered is the transition time of configuration changes. While the input parameters associated with a configuration change directly, the observations might adapt more slowly and therefore blur the expressiveness of the new segment. To prevent this from happening,

the transition part of all segments, which have been created due to configuration changes, gets truncated. If segments become smaller than a predefined threshold, they can be ignored in the upcoming phases.

3.3 Feature Extraction

Having a set of segments for each configuration, the next step is to determine the characteristics of all segments. While the literature presents multiple approaches to describe the behavior of time series, we will focus on common statistical features extracted from each segment. Nonetheless, the choice of features is not fixed, which is why any feature suitable for the individual application scenario can be used. One example for rather complex features could be the result of a kernel fitting in the context of Gaussian processes, accepting a decrease in performance. Since the goal is to capture comparable characteristics of a segment, we compute different real-valued features and combine them in a vectorized representation. In the case of COMPOSITION, we used the mean to describe the average level, the variance as a measure of fluctuation and the lower and upper quartiles as a coarse distribution-binning of values. Due to the expressiveness of features being dependent from the actual data, one possible way to optimize the selection of features is the Principal Component Analysis [6]. Simply using a large number of features to best possibly cover the variety of characteristics might have a negative influence on the measurement of dissimilarity. The reason for this is the partial consideration of irrelevant features within distance computations.

Moreover, since thresholds could be regarded as a more intuitive solution compared to additionally extracted features, this replacement would lead to a significant decrease in the number of recognized anomalies. Apart from the sensitivity to outliers, the reason is a neglect of the inherent behavior of a time series. As an example consider the measurements of an acoustic sensor attached to a motor that recently is sending fluctuating measurements, yet within the predefined tolerance. Although the recorded values are still considered as valid, the fluctuation with respect to the volume could already indicate a nearly defect motor. Finally, one initially needs to evaluate appropriate thresholds for any parameter of each configuration.

3.4 Dissimilarity Measurement

For now we discussed the exploitation of inherent information, extracted from segmented time series. The final step of GADPL is to measure the level of dissimilarity for all obtained representatives. Since no ground truth is available to define the default behavior for a specific configuration, the algorithm uses an approximation based on the given data. One problem in this regard is the variability of a default behavior, consisting of more than one pattern. Therefore, a naive approach as choosing the most occurring representative, would already fail for a time series consisting of two equally appearing patterns captured by different segments, where consequently half of the data would be detected as

Algorithm 1 GADPL

Require: Time series T , Machine parameters M , Configuration transition time p , Segment length (l_{min}, l_{max}) , Number of nearest neighbors k , Dissimilarity threshold Δ_{max}

```

 $C' = \text{cluster\_configurations}(T, M)$ 
 $R' = \{R_1, \dots, R_{|C'|}\}$ 
for all configuration segments  $C_i$  in  $C'$  do
  for all segments  $s_j$  in  $C_i$  do
     $s_j = \text{truncate\_transitions}(s_j, p)$ 
    if  $|s_j| < l_{min}$  then
       $C_i = C_i \setminus s_j$ 
    else if  $|s_j| > l_{max}$  then
       $s'_j = \text{split\_segments}(s_j, l_{max})$ 
       $C_i = C_i \cup s'_j$ 
       $C_i = C_i \setminus s_j$ 
    end if
     $R_i = R_i \cup \text{extract\_features}(s'_j)$ 
  end for
  end for
  for all configuration representatives  $R_i$  in  $R'$  do
    for all representatives  $r_j$  in  $r_i$  do
       $NN_k = \text{query\_index}(r_j, k)$ 
      if  $\Delta(r_j, NN_k) > \Delta_{max}$  then
         $\text{emit\_anomaly}(i, j)$ 
      end if
    end for
  end for

```

anomalous behavior.

As one potential solution GADPL instead uses the mean over a specified size of nearest neighbors, depicting the most similar behavior according to each segment. The idea is that even though there might multiple distinct characteristics in the data, at least a predefined number of elements represent the same behavior compared to the processed item. Otherwise, this item will even have a high average dissimilarity with respect to the most similar observations and can therefore be classified as anomaly.

Let r_i be the representative vector of the i -th segment obtained by feature extraction and let $NN_k(r_i)$ be the according set of k nearest neighbors. The dissimilarity measure Δ for r_i is defined as:

$$\Delta(r_i, NN_k(r_i)) = \frac{1}{k} \sum_{j=1}^k \delta(r_i, NN_k^j(r_i))$$

where $NN_k^j(r_i)$ corresponds to the j -th nearest neighbor and δ to a ground distance defined on \mathbb{R}^n .

Here, for the vectorized feature representations, any suitable distance function δ is applicable. In the context of COMPOSITION we decided to use the Euclidean distance for a uniform distribution of weights, applied to normalized



Fig. 1. Application of GADPL: The upper part shows the segmentation of time annotated power consumption data in percent. The lower part illustrates the result of the dissimilarity measurement, where the red rectangle indicates classified anomalies.

feature values. To further increase the performance of nearest neighbor queries, we exploited the R*-tree [1] as a high-dimensional index structure.

Given the dissimilarity for each individual representative together with a predefined anomaly threshold, GADPL finally emits potential candidates having an anomalous behavior.

4 Evaluation

In this section we will discuss the evaluation performed on a historical data set, provided in the scope of COMPOSITION. While in future, the algorithm should be applied to continuously streamed sensor data, the initial evaluation was performed on recorded data, captured over a period of seven years. The data consists of machine parameters (already classified by recipe names) and time-annotated sensor measurements including temperature value and power consumption, based on a sampling rate of 5 minutes. In addition, a separate maintenance log covers the dates of previous fan exchanges. However, malfunctions only occurred two times during runtime and are therefore comparatively rare. A confirmation of results due to actual defect components is consequently restricted to some extent. Since this project and the here presented approach are regarded as ongoing work, the outlined evaluation is continued likewise.

Figure 1 illustrates the application of GADPL, including segmentation (upper part) and dissimilarity measurement (lower part), for the time around one fan failure. Here, differently colored circles represent slices of the time series after segmentation, describing the percentage power consumption of a fan. Using the features mentioned in section 3.3, we intended to perceive deviating values and untypical fluctuations within the data, without being sensitive to outliers arising from single incorrect sensor measurements. Having one of the recorded fan

exchanges at the end of February 2012, the result of the algorithm clearly shows significantly higher values for the dissimilarity (red rectangle) prior to the event. Even though increased dissimilarity values at the end of May 2011 and around September 2011 can be explained by analyzing the original data, yet there were no recordings for a defect component at those times. However this does not automatically imply incorrect indications, since defect machine parts are not the only reasoning for anomalous characteristics in the data. An appropriate choice for the value of a maximal dissimilarity, defining the anomaly threshold, can therefore highly influence the accuracy.

Both cases of a defect fan behavior were clearly captured by the algorithm and emphasized by a high dissimilarity.

5 Conclusion

With GADPL we introduced a solution to the relevant topic of unsupervised anomaly detection in the context of configuration-based production lines. After a short outline on the topic and related work, we discussed the algorithm and the associated intention of our approach, before briefly showing the evaluation results based on the project data.

Since the approach is ongoing work, in the future we will primarily extend our evaluation based on streaming data. Although we described the algorithm using historical data, the procedure for streaming data is carried out analogous. Another point in the scope of future evaluations is the choice of more complex features and a related automated feature selection. Another idea to further improve the approach is a semantic segmentation of the time series. While currently a time series is segmented exploiting domain knowledge, a segmentation based on characteristics in the data might potentially increase the accuracy. This would also prevent from an unappropriated choice of the maximal segmentation length, which could result in a split of data within a potential motif.

Finally, we plan to investigate the correlation of anomalies within multivariate data. If GADPL in its current state is used for multivariate time series data, each dimension is processed independently. Combining inter-dimensional information within a single dissimilarity measure to cover anomalies would therefore be a useful functionality to further optimize the approach.

6 Acknowledgements

This project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 723145 - COMPOSITION. This paper reflects only the authors views and the commission is not responsible for any use that may be made of the information it contains.

References

1. Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. The r*-tree: an efficient and robust access method for points and rectangles. In *AcM Sigmod Record*, volume 19, pages 322–331. AcM, 1990.
2. Maximilian Christ, Andreas W. Kempa-Liehr, and Michael Feindt. Distributed and parallel time series feature extraction for industrial big data applications. *CoRR*, abs/1610.07717, 2016.
3. David Duvenaud, James R. Lloyd, Roger Grosse, Josh B. Tenenbaum, and Zoubin Ghahramani. Structure discovery in nonparametric regression through compositional kernel search. In Sanjoy Dasgupta and David McAllester, editors, *ICML 2013: Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *JLMR Proceedings*, pages 1166–1174. JLMR.org, June 2013.
4. Roger Grosse, Ruslan Salakhutdinov, William T. Freeman, and Joshua B. Tenenbaum. Exploiting compositionality to explore a large space of model structures. In Nando de Freitas and Kevin Murphy, editors, *Proceedings of the 28th Conference in Uncertainty in Artificial Intelligence*, Corvallis, Oregon, USA, 2012. AUAI Press.
5. James Robert Lloyd, David Duvenaud, Roger Grosse, Joshua B. Tenenbaum, and Zoubin Ghahramani. Automatic construction and natural-language description of nonparametric regression models. *CoRR*, abs/1402.4304, April 2014.
6. Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.
7. Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. Matrix profile i: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pages 1317–1322. IEEE, 2016.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





A Random Forest Based Classifier for Error Prediction of Highly Individualized Products

Gerd Gröner

Carl Zeiss Vision International GmbH
<http://www.zeiss.com>
gerd.groener@zeiss.com

Abstract. This paper presents an application of a random forest based classifier that aims at recognizing flawed products in a highly automated production environment. Within the course of this paper, some data set and application features are highlighted that make the underlying classification problem rather complex and hinders the usage of machine learning algorithms straight out-of-the-box. The findings regarding these features and how to treat the concluded challenges are highlighted in a abstracted and generalized manner.

Keywords: random forest classifier, imbalanced data, complex tree-based models, high peculiarity of data

1 Introduction

In a manufacturing process with highly individual products like ophthalmic lenses, which are produced according to personalized prescriptions, it is difficult to identify orders that are likely to fail within the production process already in advance. These products might fail due to their difficult and diverse parameter combinations. The parameters cover raw material characteristics, lens design, geometry and manufacturing parameters (i.e., machine setting values). Even such individual, prescribed products are not excluded from hard market competitions. Accordingly, avoiding waste of material and working time is an emerging problem. Obviously, since such customer-specific, individual products are not interchangeable or replaceable by other products (like in case of on-stock products), it is highly valuable to avoid any kind of scrap / failure already beforehand the production. Summing up, it is becoming more and more useful to analyze product (order) parameters and find features and feature correlations in order to predict (potential) failures already prior to the start of any manufacturing process.

In our case, we are confronted with a rather hard problem since the products can not be perfectly discriminated into good or bad ones solely based on their product characteristics (which are given by individual prescription and design in our case) and their corresponding target processing parameters. Therefore, it is a challenging machine learning (ML) task to remedy this problem within an advance distinction between good and potential faulty products, while, at the

same time, avoiding ML pitfalls like over-fitting. Furthermore, the pure number of features is high and the data set is quite imbalanced, hampering the straight forward exploitation of ML models.

Until now, ML is used for error detection in different manufacturing areas (e.g., [1–3]), but due to the domain-specific data (highly individualized) and fully-automated and very standardized manufacturing processes, the gap between different parameter combinations and the resulting processing steps is an open challenge for applying ML technologies and assessing their benefits accordingly.

We present a *random forest classifier* for error prediction that resulted from a deep analysis of different ML algorithms, which has been used to train various models. These models are evaluated in terms of their classification quality. The best model is presented in detail. Interestingly, doubts (like difficult distinction) and findings (like important features) of the domain experts form the manufacturing division were confirmed by the model. Finally, we give an argumentation why the random forest model outperforms other (rather complex) models like Neural Networks and Support Vector Machines (SVM) within this particular use case.

2 Background

This section shortly outlines background information on a particular studied use case, followed by some principles on machine learning.

I. Use Case: Error Recognition and Prediction. For an ordered product, we focus on the relevant product features and the according machine setting parameters. Summing up to 130 features that describe the product, i.e., lens in our case by data on geometry, shape, target prescriptions, coatings and tinting values. We removed identifiers like order number and dates. In the used data set, we have about 560000 entries in total (i.e., products), covering those products without errors and such cases, where the first production was erroneous and a further (second) production cycle was necessary.

As we train, test and evaluate our model with historical data, for each product there is the corresponding characteristic whether it is an error or a non-error (binary classification). Since we are interested in an advance classification of products (and their corresponding to-be processing parameters), we neglect in the historic data those errors that were caused by operators, by unexpected machine failures or by other arbitrary circumstances. The remaining proportion of (final) errors is about 5.4 %.

II. Machine Learning (in Practice). Based on the use case, we are faced with a binary classification problem (i.e., we distinguish – at least in a first step – between good and potential bad products). This problem (*classification*) constitutes one group of algorithms in the realm of supervised machine learning, while the second group of algorithms of supervised learning is referred to as a *regression* problem, where instead of discrete categories (as in our case) a continuous value is the target output of a model. Among *classifications*, there are

a variety of algorithms (cf. [4–6]), ranging from rather basic ones like regression and Naive Bayes, to more difficult algorithms (in terms of setting-up and computation) like artificial neural networks (ANN), support vector machines (SVM), decision trees and extensions of them like random forests classifiers (RFCs) and boosted decision trees. Boosted decision trees and random forests belong to the so-called ensemble algorithms, i.e., a set of trees or a forest is built by an ensemble of decision trees. Ensemble algorithms implement methods to generate multiple classifiers and then aggregate their results (cf. [16]). Boosted decision tree algorithms apply a strategy of state-wise optimization of trees (measured in terms of loss functions) [14, 15]. Trees within the ensemble of random forests are built by randomly selecting the input features. Each tree in the ensemble is obtained by randomly selecting the input features. Within each tree, each node is still split by using the best feature (measured in terms of cost functions). The final result of the forest is obtained by unit votes from the trees for the most popular class.

3 Characteristics of the Data Set and the Application Scope

The data set is obtained from a rather dedicated domain, following a production process for highly individualized products, there are some essential key characteristics that are comparable and transferable to different problems in completely other domains. Therefore, we have to tackle challenges to cope with the following data and application characteristics.

The data set is highly *imbalanced*, which is actually in the nature of error and non-error classification problems. As already mentioned, we have a relationship of roughly 5.4 % belonging to the minority class (error case), while slightly more than the remaining 94.6 % of the data samples belong to the majority class (non-error case). It is well known that the best classification results can be achieved on balanced data sets (cf. [11–13]). Furthermore, in our case, we are not only interested in the correct classification, we also want to know which are the most influential features for ending-up in one of these two classes. Thus, a sound prediction model that is able to do a proper classification (i.e., a non-guessing solution!) is needed.

A further property is the *complexity* of the model. The pure number of samples (roughly 560000 entries in the data set) is a decent size, but the compared amount of features (about 130) is rather high. In particular, not only the number itself is an issue, it is rather the feature characteristic that counts for complexity, as we will see later. There are no dominating single features and the number of influential features is high, ending up with models that need a deep consideration of feature manifestation and combinations, as demonstrated in the next section.

Finally, the third characteristic is the vague *discriminability*, which is the most difficult one to handle in our case. Given all the features of a particularly ordered product of an error case, the manufacturing process at the first time has failed, while the second run with quite similar or even the same features (in-

cluding machine setting parameters) ended-up with a good quality. Accordingly, such a concrete characteristic of product attributes is not able to determine in advance whether an error or a non-error case is given.

4 A Random Forest Model for Error Prediction

This section presents the set-up of the model training, starting with the necessary data preparation steps, the part of algorithm set-up and result comparison, followed by the evaluation and an discussion of the design decisions and the achieved results.

4.1 Data Preparation and Preprocessing

After the basic step of creating a data model within a database and cleaning tasks like dealing with outliers and missing values, we applied several feature engineering steps. We have to deal with various categorical values. Even if some algorithms are able to directly handle them, we applied a general encoding of all categorical features. We use the established `one-hot-encoding` method for this step. Furthermore, for some parameters with different values within the production steps (steps in the production process), the results improved by adding aggregations of these parameters like average values to the data set.

4.2 Features and Feature Distribution

Among the features (independent variables) there is a clear ordering regarding feature importance, but there is no clear dominance of a single feature or of a small group of features. For instance, the relative importance of the most important feature is about 0.0383, the 10th important feature still reaches a relative importance of roughly 0.0302.

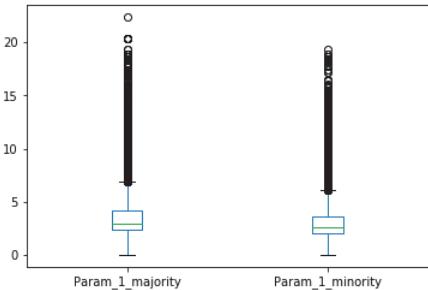
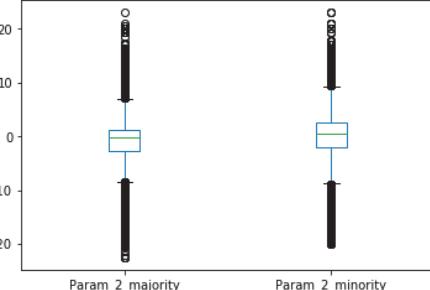
Figure 3 shows the distribution of the first and the tenth important feature. The features are renamed here, param. 1 refers to the first / most important feature (Figure 1) and param. 2 to the tenth important feature (Figure 2). We added suffixes in the plots to show the distribution of the error and non-error case separately. The plots depict the distribution of the whole data set (i.e., including data of the train and test part). The left box (i.e., the suffix “majority”) refers to the values of the majority class (i.e., non-error case), while the suffix “minority” refers to the values of the minority class (i.e., error case)).

4.3 Algorithm Comparison and Selection

We built all models by training with several algorithms, using the Python programming language and libraries like the Scikit-learn library¹ in Python.

The data set is split up into training (0.7) and test (0.3) data. The results show that the data contains rather complex interactions among the most relevant

¹ Scikit-learn: <http://scikit-learn.org/stable/>

**Fig. 1.** Most important feature.**Fig. 2.** 10th most important feature.**Fig. 3.** Box plots for the distribution of two features.

features. Moreover, the discrimination between error and non-error (if possible at all) requires the comprehensive consideration of various features and their relations, which has been outlined in our comparison. For instance, less-complex algorithms like Naive Bayes and regressions are not able to do a decent classification. Algorithms known as complex and partially hard to initialize like support vector machines (SVM) and artificial neural networks (ANN) are able to make proper binary classifications, but with a low F1 score. Tree-based algorithms outperform all others. The best results are obtained by boosted trees and, slightly better, by random forest classifiers.

Table 1 shows an excerpt of an algorithm comparison. The first column describes the used algorithm to train the model. Column two gives the setting parameters of the algorithm. If no parameter is given, the default values are taken (from Scikit learn). The presented setting parameters are those which ended up in the best results, mainly received by several trials and applying cross-validation strategies (We used a 5-fold cross validation on the training data set).

The third column describes the performance in terms of *precision*, followed by the *recall* in column four and the summarized *F1 score* in column five, concluded by the ROC-AUC value (area under the ROC curve). All models were trained with these algorithms from the Scikit learn package in Python.

For the *random forest classifier (RFC)*, we explicitly parametrized the algorithm with the minimum number of samples for a split to 3, and no limit of the maximum depth of the branches in a tree. The quality of a split is measured by the Gini impurity. This measure judges the quality of a selected target variable, which is used to split a node, i.e., reflecting the *importance* or “*best split criteria*” in a tree. The Gini impurity measures how often an element is wrongly classified (i.e., assigned to a subset (bin)), if the “correct” label reflects the random label assignment of the distribution of labels within the subset.

The *boosted decision tree* (implemented by AdaBoost in Scikit learn) has been constituted within a rather similar setting. The tree properties are set to the minimum number of samples for a split to three, no limitation on the depth

and also the Gini impurity is used to assess the split quality. The learning rate shrinks the contribution of a single classifier within the ensemble. We use the default boosting algorithm (SAMME.R), which aims at converging faster than the other options.

The *artificial neural network (ANN)* (also referred to as multi-layer perceptron - MLP - classifier) uses an adaptive learning rate, which means that the learning rate is reduced (divided by five) as far as in two successive runs the training loss does not decrease. The parameter alpha represents the regulation of the L2 penalty (i.e., Ridge penalty). The value is higher than the default, implying smaller coefficients (weights). The parameter on the hidden layers defines the number of hidden layers (five in our case) and also the number of nodes (neurons) in each layer.

For the *support vector machines (SVM)* (or support vector classifier), we use the rbf (radial basis function) kernel. (The rbf kernel uses a squared Euclidean distance as measurement for data (point) separation. The gamma coefficient is set to auto, which means that the quotient from one and the number (n) of features. The penalty parameter for errors (C) is five. This parameter is balancing between errors in training compared to errors in testing, i.e., it influences the generalization of a classifier to unseen data.

Table 1. Comparison of Model Performance.

Algorithm	Parameter	Performance			
		Precision	Recall	F1 Score	ROC-AUC
RFC	criterion: Gini min-sample-split: 3 max-depth (tree): none	0.74	0.4	0.52	0.72
Boosted Tree (AdaBoost)	criterion: Gini min-sample-split: 3 max-depth (tree): none learning rate: 0.4	0.72	0.39	0.51	0.71
ANN (MLP)	learning rate: adaptive alpha (L2 penalty): 0.1 hidden layer sizes: (70,70,50,40,40)	0.59	0.24	0.34	0.55
SVM	kernel: rbf gamma (coef.): auto (=1/n) C (penalty for error): 5%	0.55	0.19	0.28	0.52

The random forest classifier was set up by using a 5-fold cross validation (grid search with parameter alternatives) in order to find the best parameter combinations (e.g., the minimum samples within a leaf). We need very deep trees (setting no depth limitation) and a very low splitting rate in the nodes (best results are achieved with three sample splits). The average tree depth is 51. A further interesting finding is the distance between precision and recall. While the precision is about 0.74, recall ended up with 0.4 (F1 score is 0.52).

Fig. 4 depicts the ROC curve (Receiver-Operating-Characteristic curve) for the random forest classifier. The true positive rate (i.e., the recall rate or also referred to as sensitivity) is depicted on the y-axis, the x-axis shows the false positive rate.

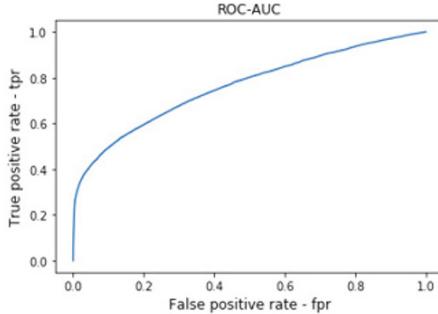


Fig. 4. The ROC curve of the random forest classifier.

4.4 Algorithm Comparison and Selection

While it is often argued that both described tree algorithms (i.e., boosted decision trees and random forests) tend to perfectly adapt their feature values and thus suffer often from overfitting, Breimann [5] showed that random forests are robust against overfitting, providing (among others) possibilities to set regularization parameters.

4.5 Evaluation, Results and Design Decision Revisited

It is worth to notice that due to the rather low ratio of the error samples (so-called minority class), we applied re-sampling methods [7, 8] to obtain a more balanced data set. The best results were achieved by down-sampling (i.e., reducing the data set size) in combination with a slight up-sampling, such that the error ratio raises up to nearly 18 %. There is no dominating feature among the most important features.

While several practical comparisons (e.g., [19]) show that the complex ANN outperforms random forests, the variety of important (but not dominating features) combined with their different results of interactions and the threat of overfitting might cause the predominance of random forests in our case.

Nevertheless, we stress that the best results of the random forests is based on the underlying data set and application use case with no indication as a general superiority of random forest classifiers to other classification algorithms, which was for instance argued in [18], but later contradicted (in terms of generalizability) in [17].

It is definitely hard (or even impossible) to explain why a certain algorithm (like random forests in our case) provide the best results compared to other algorithm. We will follow some discussions like on KDnuggets², on blocks like Towards Data Science³ as well as in a work on energy consumption analysis [19].

The models built by random forests are known as rather robust models, i.e., they are able to better handle outliers, missing data or just weird values. We realize a slight overfitting, which is a well-known problem of random forests (especially with deep trees), but it is minor and negligible in our case.

Neural networks (and also SVM) are more difficult to parametrize. Although we applied various training iterations with different parameter settings (always including default parameters), it is still imaginable that a better parameter combination for the algorithms exists and the resulting model would outperform our current best random forest solution. Furthermore, our model covers very complex interactions among features, which is shown by the very deep trees (compared to the total number of features). However, all features are numerical values or categorical values, there are no images and we are not in the realm of image or speech processing, which are known areas where neural networks and SVM (especially for text data) mostly outperform other algorithms.

5 Summary and Outlook

In this paper, we presented a study for in-advance error classification in a highly individualized production environment. The best predictions are achieved by tree-based algorithm, in particular by a random forest classifier that achieves a rather decent precision rate to forecast whether a particular ordered product is likely to fail or not. However, the recall is comparable low. As the data set is highly imbalanced, we used sampling strategies to slightly improve the ratio between errors and non-errors in our data set.

As future work, we train our models with an updated (newer) data set, containing more data in both dimensions, i.e., for data entities samples, but also slightly more features. The expectation is that this will increase the algorithm performance.

References

1. Henmi, T., Deng, M., Yoshinaga, S.: Early Detection of Plant Faults by Using Machine Learning. *Int. Conf. on Advanced Mechatronic Systems (ICAMechS)*, 2016
2. Zidek, K., Maxim, V.: Diagnostics of Product Defects by Clustering and Machine Learning Classification Algorithm. *Journal of Automation and Control*, vol.3, 2015

² Post on KDnuggets: “When Does Deep Learning Work Better Than SVMs or Random Forests?”, <https://www.kdnuggets.com/2016/04/deep-learning-vs-svm-random-forest.html>

³ <https://towardsdatascience.com>

3. Meshram, A., Haas, C.: Anomaly Detection in Industrial Networks using Machine Learning: A Roadmap. *Machine Learning for Cyber Physical Systems: Selected papers from the International Conference ML4CPS 2016*. Ed.: J. Beyerer, Springer, pp. 65–72, 2017
4. Géron, A.: Hands-On Machine Learning with Scikit-Learn & TensorFlow. O'Reilly, 2017
5. Breiman, L.: Random Forests. *Machine Learning*, pp. 5–32, vol. 7, Kluwer Academic Publishers, 2001
6. Rashid, T., Neuronale Netze selbst programmieren. O'Reilly, 2017
7. Lemaitre, G., Nogueira, F., Aridas, C.K.: Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research*, vol. 18, pp. 1-5, 2017
8. Batista, G.E.A.P.A., Prati, R.C., Monard, M.C.: A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. *ACM SIGKDD*, vol. 6 (1), pp. 20–29, 2004
9. Ahmad, M.W., Mourshed, M., Rezgui, Y.: Trees vs Neurons: Comparison between Random Forest and ANN for High-Resolution Prediction of Building Energy Consumption. *Energy and Buildings*, Elsevier, vol. 147, pp. 77–89, 2017
10. Zhang, Y., Guo, W., Ray, S.: On the Consistency of Feature Selection With Lasso for Non-linear Targets. *Proc. of the 33rd Int. Conference on Machine Learning*, vol. 48, pp. 183–191, 2016
11. Eitrich, T., Kless, A., Druska, C., Meyer, W., Grotendorst, J.: Classification of Highly Unbalanced CYP450 Data of Drugs Using Cost Sensitive Machine Learning Techniques. *Journal of Chemical Information and Modeling*, vol. 47 (1), pp. 92–103, 2007
12. Wang, S., Yao, X.: Multiclass Imbalance Problems: Analysis and Potential Solutions. *Systems Man Cybernetics Part B - Journal IEEE Transactions on Cybernetics*, vol. 42, pp. 1119–1130, 2012
13. Kubat, M., Matwin, S.: Addressing the Course of Imbalanced Training Sets: One-Sided Selection. *Proc. of the 14th Int. Conference on Machine Learning*, pp. 217–225, 1997
14. Wyner, A.J., Olson, M., Bleich, J., Mease, D.: Explaining the Success of AdaBoost and Random Forests as Interpolating Classifiers. *Journal of Machine Learning Research*, vol. 18, pp. 48:1–48:33, 2017
15. Friedman, J.: Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics*, pp. 1189–1232, 2001
16. Liaw, A., Wiener, M.: Classification and Regression by Randomforest. *R news*, vol. 2 (3), pp. 18–22, 2002
17. Wainberg, M., Alipanahi, B., Frey, B.J.: Are Random Forests Truly the Best Classifiers?. *Journal of Machine Learning Research*, vol. 17, pp. 110:1–110:5, 2016
18. Fernández-Delgado, M., Cernadas, E., Barro, S., Amorim, D.: Do we Need Hundreds of Classifiers to Solve Real World Classification Problems? *Journal of Machine Learning Research*, vol. 15, pp. 3133–3181, 2014
19. Ahmad, W. M., Mourshed, M., Rezgui, Y.: Trees vs Neurons: Comparison between random forest and ANN for high-resolution prediction of building energy consumption *Journal on Energy and Buildings*, vol. 147, pp. 77–89, 2017

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Web-based Machine Learning Platform for Condition-Monitoring

Thomas Bernard¹, Christian Kühnert¹, Enrique Campbell²

¹ Fraunhofer Institute for Optronics, System Technologies and Image Exploitation IOSB
Karlsruhe, Germany

² Berliner Wasserbetriebe, Neue Jüdenstraße 1, Berlin

* Corresponding author. Tel.: +49-721-6091-360

E-mail address: thomas.bernard@iosb.fraunhofer.de

Abstract. Modern water system infrastructures are equipped with a large amount of sensors. In recent years machine-learning (ML) algorithms became a promising option for data analysis. However, currently ML algorithms are not frequently used in real-world applications. One reason is the costly and time-consuming integration and maintenance of ML algorithms by data scientists. To overcome this challenge, this paper proposes a generic, adaptable platform for real-time data analysis in water distribution networks. The architecture of the platform allows to connect to different types of data sources, to process its measurements in real-time with and without ML algorithms and finally pushing the results to different sinks, like a database or a web-interface. This is achieved by a modular, plugin based software architecture of the platform. As a use-case, a data-driven anomaly detection algorithm is used to monitor the water quality of several water treatment plants of the city of Berlin.

Keywords: Machine-learning; water quality monitoring; anomaly detection; plugin architecture; data fusion.

1 Introduction

In recent years, a large number of new water quality and hydraulic sensors in water distribution networks and water treatment plants have been installed. Reasons for this trend are (1) a lot of new sensor companies and corresponding new sensors appeared on the market which means decreasing costs and increasing performance of the sensor units; (2) due to wireless communication technologies (e.g. GSM) the installation costs are drastically decreasing. Hence, there is a need for the development of integrated platforms for the storage, visualisation and enhanced data analysis of these data. The benefit of advanced data analysis in water infrastructures has been already investigated for different scenarios, e.g. monitoring of drinking water quality 4, forecasting of the water consumption 6 or the modelling of sediment transport 1. However, different data suppliers and old plants containing an outdated IT-infrastructure still complicate the integration of state-of-the-art data analysis algorithms. In spite of the fact that many

IoT and data analysis platforms are available nowadays the effort for the integration of these platforms in the IT infrastructure of water utilities and the implementation of ML algorithms is still very high. To overcome some of these challenges, this paper presents a generic data fusion and analysis platform with the focus on condition monitoring of the WDN with machine learning algorithms. The platform follows a plug-in based architecture, which means that depending on the specific needs of the current use case (e.g. saving data in a database, performing anomaly detection) different software components can be installed. As a use case, the platform is used to perform the condition-monitoring of nine water quality measuring stations in parallel with a combination of Principal Component Analysis (PCA) 2 and Gaussian Mixture Models (GMMs) 9. The results of the machine learning algorithms, comprising the learned process map, the state trajectory and the anomaly index, are visualized for all stations in a web-interface.

2 Platform Architecture

The architecture of the proposed platform consists in three main parts shown in figure 1: (1) the platform core, (2) a plugin structure and (3) a web-interface. The platform core is responsible for the management of the different software modules and data handling and described in section 2.1; the plugins provide the required use case specific application functionality (e.g. analysis algorithms; connection to data source) and are described in section 2.2. Finally, the web-interface, used to give a feed-back to the user, is explained in section 2.3.

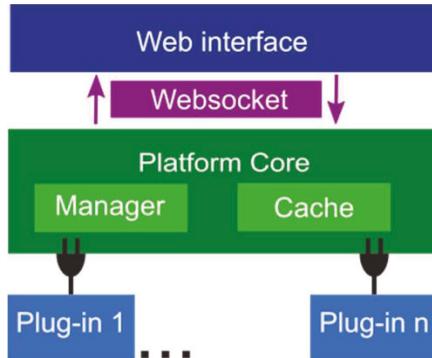


Fig 1: Plug-in architecture of the platform for real-time data analysis applications

2.1 Platform Core

The platform core's purpose is to provide the stability to allow communication between all components - no matter their purpose, data rate or lifetime. Its main purpose is to act as information hub providing a standard interface for all plugins. Therefore, the platform core utilizes the mediator design pattern 3 to decouple all plugins from each other. The resulting communication topology of plugins and core is a star network with

the core as central component, thus preventing any plugin to plugin communication. The core itself uses the *Model-View-Controller* (MVC) pattern 3.

The *core manager* is the controller of the platform. It is the owner of all plugins as well as the core cache and responsible for their creation and destruction. Since it is also the facade for the whole core, it is known by reference by all plugins, which need to request access for each core cache entry they want to access.

The *core cache* acts as model to separate the core's data from its logic. In order to establish either a read only or read/write connection to the core cache, a plugin has to be granted permission by the core logic. Once a connection is established, the plugin receives a local copy of the requested core cache data which stays in sync with the cache via the observer pattern 3.

2.2 Plugins

To maintain the maximum amount of flexibility, the platform follows a plugin based architecture. This means that depending on the specific needs of the current use case different software components can be integrated into the platform. Basically, a plugin represents a software module fulfilling a specific task. Examples are the connection to the SCADA system of the water utility; the implementation of an event detection algorithm or the automated generation of a daily, weekly or monthly report. Plugins employ the factory pattern 3 to allow creating several instances which can be configured started and stopped individually.

2.3 Web interface

A web interface is provided to offer a cross device interface for different operating systems to access and interpret the data. Therefore, the main aim of the interface is to provide the users a quick overview of the results of the data analysis algorithms. Since it is implemented as a homepage, it can be accessed with any device with an internet connection from anywhere from multiple concurrent clients. Data is transferred to the web-client by using web sockets.

3 Data-driven Condition-Monitoring

In literature numerousness approaches for data-driven condition-monitoring have been proposed. Among them, 10 or 11 provide good overviews of this topic. The in this paper used method for data-driven condition-monitoring of the measuring stations is covers several steps and is sketched in **Fig 2**. Initially, a z-score normalization 2 of the measurements is performed. Next, the initial data is reduced down to two dimensions using as principal component analysis (PCA) 8. Finally, using the first two principal components, a Gaussian Mixture model 9 is used for the detection of anomalies. All steps are described in the following sections

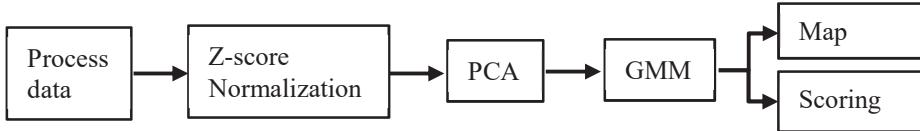


Fig 2: Work-flow for data-driven condition-monitoring on measurement stations

3.1 Z-score normalization

It is assumed that $x[k] \in \mathbb{R}$ with $k = 1 \dots K$ is the time series of a process variable with mean value μ and standard deviation σ . Hence, the set of all process variables is described as

$$X = [x_1[k], x_2[k], \dots, x_p[k]] \quad (1)$$

With p being the number of process variables resulting in the matrix $X \in \mathbb{R}^{(K \times P)}$. Finally, the z-score normalization is defined as

$$Z = \frac{x_j - \mu_j}{\sigma_j} \quad (2)$$

With $j = 1 \dots P$. As mentioned, the PCA is calculated using the matrix Z containing the normalized process variables.

3.2 Principal Component Analysis

The principal component analysis (PCA) is a procedure of multivariate statistics to structure large data sets. In that case it is used for model reduction. The main concept is to perform an orthogonal transformation to map the set of correlated variables into a set of linear, uncorrelated ones. Mathematically, the principal components then cover the variance accounted for in the data set. The calculation of the principal components is carried out by computing the eigenvectors of the covariance matrix being defined as:

$$\Sigma = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \dots & \sigma_{1p}^2 \\ \sigma_{12}^2 & \sigma_{22}^2 & \dots & \sigma_{2p}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1p}^2 & \sigma_{2p}^2 & \dots & \sigma_{pp}^2 \end{bmatrix} \quad (3)$$

with σ_{ij}^2 being the covariance of the two standardized variables $z_i[k]$ and $z_j[k]$ in the variable set. Next, the eigenvalues λ of the covariance matrix are calculated and sorted in ascending order. This results in the final diagonal matrix $\Lambda \in \mathbb{R}^{P \times P}$ defined as

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_p \end{bmatrix} \text{ with } \lambda_1 \geq \dots \geq \lambda_p \quad (4)$$

In a next step, the corresponding eigenvectors of the eigenvalue matrix Λ are calculated and summarized in columns. This results in the matrix $\Gamma \in \mathbb{R}^{P \times P}$

$$\boldsymbol{\Gamma} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1p} \\ \gamma_{21} & \gamma_{22} & \cdots & \gamma_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{p1} & \gamma_{p2} & \cdots & \lambda_{pp} \end{bmatrix} \quad (5)$$

Finally, the matrix $\boldsymbol{\Gamma}$ is used to perform the linear transformation $\mathbf{Z} \rightarrow \mathbf{Y} = \boldsymbol{\Gamma}^T \mathbf{Z}$, while \mathbf{Y} contains the principal components. For example, $y_1[k] = \gamma_{11} z_1[k] + \cdots + \gamma_{p1} z_p[k]$ corresponds to the first principal component.

3.3 Gaussian Mixture Models

A Gaussian Mixture Model (GMM) is a parametric statistical model, which assumes that the data comes from several Gaussian sources. In detail, a GMM is defined as:

$$p(x|\Theta) = \sum_{i=1}^K \omega_i p_i(x|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \quad (6)$$

With K being the number of density components, ω_i , with $\omega_i \geq 0$ and $\sum_{i=1}^K \omega_i = 1$, the mixture weight and $p_i(x|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ the individual Gaussian distributions being defined as

$$p_i(x|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \frac{1}{(2\pi)^{p/2} |\boldsymbol{\Sigma}_i|^{1/2}} e^{-\frac{1}{2}[(x-\boldsymbol{\mu}_i)' \boldsymbol{\Sigma}_i^{-1} (x-\boldsymbol{\mu}_i)]} \quad (7)$$

with $\boldsymbol{\mu}_i$ the mean vector and $\boldsymbol{\Sigma}_i$ the covariance matrix. The log-probability of a sample $x \in \mathbb{R}^{1 \times P}$ is then determined as

$$\hat{a} = \sum_{p=1}^P \log \sum_{i=1}^K \omega_i p(x|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \quad (8)$$

with $\hat{a} \in \mathbb{R}$. The training of the GMM means to estimate the weights ω_i , the mean $\boldsymbol{\mu}_i$ and the covariance $\boldsymbol{\Sigma}_i$. Therefore, an usually an Expectation Maximization (EM) algorithm is used [9]. The EM algorithms tries to increase the expected log-likelihood of the complete training data set by iteratively changing the GMM parameters until they converged. In this paper, for training the GMM, the first two principal components from the initial training set are used.

3.4 Process mapping and trajectory

A process map of a measuring station from Berliner Wasserbetriebe is shown in Fig 3. For the generation of the process map, the x-axis represents the first, the y-axis the second principal component. The trained Gaussian Mixture Model is visualized in terms of isobars, while red represents a cluster center and blue areas without data. New measurements are transferred into principal component space and, using the first two components, is mapped into the process map. If the measurements are mapped into the blue area, this indicates a possible anomaly. Fig 3 on the right side shows an example of an anomaly, resulting from a sudden reduction of the redox-potential at one of the measuring stations in Berlin. The trajectory is moving away from the GMM cluster center.

Finally, the log-probability from the GMM for a measurement can be used as anomaly index which defines if a system is running in normal or abnormal state. A low value of

$\hat{\alpha}$ indicates a not normal state, while a good practice for a threshold selection is to take the lowest value of $\hat{\alpha}$ resulting from the training data.

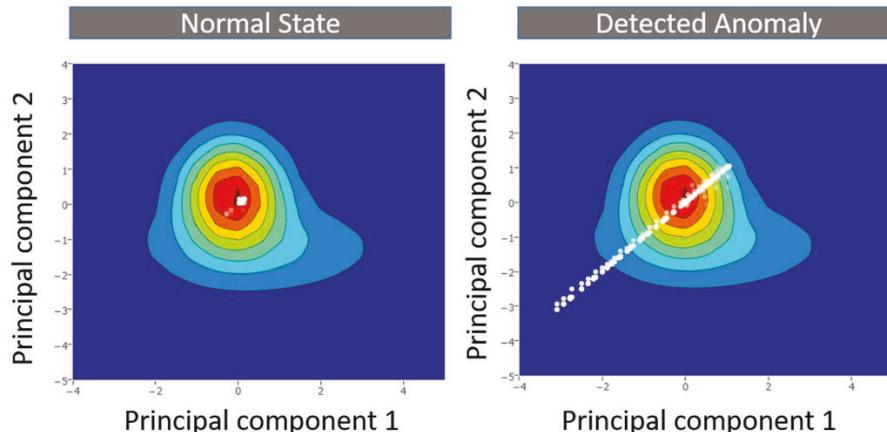


Fig 3: (Left) Visualization of the calculated GMM and the trajectory of a measuring station from Berliner Wasserbetriebe in normal state. (Right) The same map with a detected anomaly, namely a reduction of the redox-potential in the measurements.

4 Use case: Water quality Monitoring of Water Treatment Plants in Berlin

Within the French-German research project ResiWater 7 a monitoring of the water quality parameters of nine water treatment plants of the city of Berlin has been built up. At each water treatment plant the parameters pH, turbidity, redoxpotential, oxygen and conductivity are measured. The analysis chain consists in these steps: (1) Data fetching from BWB's SCADA system and storing in a local database for analyses, (2) using the in section 3 described data-driven condition-monitoring algorithm for each monitoring station, (3) generate graphs compromising the results of the condition-monitoring system over the last couple weeks; (4) pushing results to a web-client for visualization and interpretation of the event. All developed plugins are briefly described in the following section.

4.1 Plugins

For the use case of water quality monitoring, the following plugins are implemented.

- *Data polling and parsing plugin (1):* The measurements from the water quality monitoring stations are exported by the SCADA system as chunked .csv files on a secure FTPS server with a sample time of a few minutes. A plugin cyclically polls to the FTPS server and checks if new data is available. In this case the corresponding files are downloaded, parsed and written into the cache. From the cache, they are analyzed by the condition-monitoring plugin.

- *Condition-Monitoring plugin (2):* The in section 3 described approach for data-driven condition monitoring has been implemented in this plug-in. For each measuring station (in total nine stations are monitored), data representing the normal state has been selected and used for training the PCA and GMM. New acquired sensor data is evaluated by the event detection module. If the log-probability for a new measurement is below the predefined threshold, an alarm is raised by the plug-in which sends this information to the platform cache.
- *Graph generation plugin (3):* This plugin generates graphics containing the results of the event detection modules as well as the corresponding measurements. These graphics can be accessed from the web-client and provide a long term overview of the detected events in the network.
- *Realtime-web plugin (4):* This plugin pushes the online-measurements as well as the current results of the event detection module via web sockets to the web-clients. To avoid too much network traffic, values are only pushed on change and not on a fixed time stamp.

Fig 4, upper side, shows the plug-in manager with the loaded plug-ins. The lower plot gives a screenshot of the real-time data cache containing results from the different plug-ins

NAME	DESCRIPTION	KEYID
ResiWater Eurometropole cyclic csv importer	Cyclically checks if new data arrived and imports into	a3cd6219-65ef-4e3d-b674-dcca03dda276
ResiWater Eurometropole cyclic graphic generator	Cyclically checks if new plots can be generated from	c5b13505a-ce9f-437b-b8c5-2a52483fb787
ResiWater DB To Cache Writer Plugin	Cyclically pushes new values from the dB to Cache	cf656f61-2191-4923-b1d9-7a8df495b749
ResiWater PCA-GMM Monitoring	Cyclically evaluates new data in dB	cbba61f-2b6e-482c-b687-d2f7a92b8ae5
SignalRworker	Pushes messages on SignalHub	660ad5f97-cf21-4e3d-b674-dcca03dda276

Fig 4: (Upper plot) Plug-in manager with loaded plug-ins for monitoring; (lower plot) real-time data cache

4.2 Web-interface

The web-interface provides an overview of the current state of the monitored measurement stations, the process map with the trajectory, as well as information about the historic results from the condition-monitoring algorithms. Furthermore, the complete

website is kept responsive, which means that the results can be visualized on a tablet or smartphone as well. In summary the interface covers the following main features:

- *Dashboard*: The dashboard consists of a set of tiles and deals as a summary of the current states of each measuring station. Basically, tiles can be in green (normal state) and in red color (anomaly detected), depending on the value of the anomaly index (see section 3.3). If the index falls below a predefined threshold, the color changes from green to red. A screenshot of the dashboard is shown in Fig 5 left side.
- *Process map and trajectory visualization*: The calculated map as well as the trajectory and the anomaly index, described in section 3, are visualized in the web-client. This gives an overview of the current state of the process and shows if it is in normal or abnormal state. A screenshot of the process map is given in Fig 6.
- *Time series visualization*: The web-client provides the possibility to give historic and real-time access to the anomaly indices (Fig 5 middle). Additionally, in a pre-defined time-frame, a plot of the alarm index with the corresponding measurements is generated. A screenshot is shown in Fig 5 on the right hand side.



Fig 5: (Left) Screenshot of the Dashboard; (middle) exemplary anomaly indices for measuring stations; (right) graph covering GMM scoring results with the corresponding measurements of the last month for a measuring station

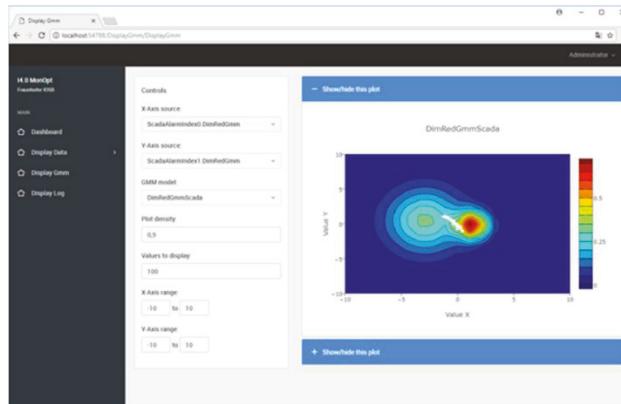


Fig 6: Visualization of the process map and trajectory within the web-client

5 Conclusion

This paper presents a generic platform for data analysis with a focus on data-driven condition-monitoring in water distribution. Therefore, a plugin based software architecture is proposed, which can be used to collect data from different sources, treat data with different analysis algorithms and provide the results by a web-based user interface. Due to the plugin structure, the platform provides a large flexibility and can be adapted for very complex scenarios. For data analyses, a data-driven condition-monitoring approach based on a combination of Principal Component Analysis and Gaussian Mixture Models was realized. Within this approach, the original input data is reduced down to two dimension to generate a map of the process. Next, this map is used in combination with the calculated process trajectory to visualize if the process is close to a cluster center, meaning in a normal state. Furthermore, an anomaly index is calculated, which defines if the process is in normal or abnormal state. As a use-case, the results of the monitoring of the water quality parameters in the city of Berlin has been presented.

Acknowledgements

The project ResiWater [7] is supported by the German Federal Ministry of Education and Research (BMBF) and by the French Agence Nationale de la Recherche (ANR).

References

1. B. Bhattacharya, R.K. Price, D.P. Solomatine: Machine Learning Approach to Modeling Sediment Transport, *Journal of Hydraulic Engineering*, 2007
2. C- Bishop: Pattern recognition and machine learning, Springer, 2006
3. E. Gamma, R. Helm, R. Johnson, J. Vlissidis: Design Patterns: Elements of Reusable Object-oriented Software, Addison-Wesley, 1994
4. C. Kuehnert et. al.: A new alarm generation concept for water distribution networks based on machine learning algorithms, 11th International Conference on Hydroinformatics, 2014
5. T. Marwala: Gaussian Mixture Models and Hidden Markov Models for Condition Monitoring, In:Condition Monitoring Using Computational Intelligence Methods, Springer, 2012
6. Z. Ren: Short-term demand forecasting for distributed water supply networks: A multi-scale approach, WCICA, 2016
7. Project ResiWater - Innovative Secure Sensor Networks and Model-based Assessment Tools for Increased Resilience of Water Infrastructure, project website: <https://www.resiwater.eu>; funded by BMBF (13M13688) and ANR (ANR-14-PICS-0003)
8. Fodor, Imola K. A survey of dimension reduction techniques. No. UCRL-ID-148494. Lawrence Livermore National Lab., CA (US), 2002.
9. Reynolds, Douglas. "Gaussian mixture models." *Encyclopedia of biometrics* (2015): 827-832.
10. Qin, S. Joe. "Survey on data-driven industrial process monitoring and diagnosis." *Annual reviews in control* 36.2 (2012): 220-234.
11. Yin, Shen, et al. "A review on basic data-driven approaches for industrial process monitoring." *IEEE Transactions on Industrial Electronics* 61.11 (2014): 6418-6428

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Selection and Application of Machine Learning-Algorithms in Production Quality

Jonathan Krauß¹, Maik Frye¹, Gustavo Teodoro Döhler Beck¹, Robert H. Schmitt²

¹ Fraunhofer Institute for Production Technology IPT
Steinbachstr. 17, Aachen 52074, Germany

{jonathan.krauss,maik.frye,gustavo.beck}@ipt.fraunhofer.de

² Laboratory for Machine Tools WZL RWTH Aachen University
Steinbachstr. 19, Aachen 52074, Germany
robert.schmitt@rwth-aachen.de

Abstract. Due to the increase in digitalization Machine Learning (ML)-algorithms bare high potentials for process optimization in the production quality-domain. Nowadays, ML-algorithms are hardly implemented in the production environment. In this paper, we present a tangible use case in which ML-algorithms are applied for predicting the quality of products in a process chain and present the lessons learned we extracted from the application. In the described project, the process of choosing ML-algorithms was a bottleneck. Therefore we describe a promising approach how a decision making tool can help selecting ML-algorithms problem-specifically.

1 Data-Driven Modeling in the Production Quality

Digitalization has led to a steady increase in data in recent years. Through higher computing power, it is possible to process the large amount of data [1]. Analyzing the acquired data can enhance both the understanding and the process efficiency - or to describe it in the words of Peter Sondergaard: "Information is the oil of the 21st century, and analytics is the combustion engine" [2]. Especially sectors like the financing-domain or the marketing-domain are leading when it comes to generate value from data [3]. In particular, the use of Machine Learning (ML)-algorithms increased over the last decade. The main reasons for this trend, apart from the higher computing power and data input mentioned above, are the increasing reliability of the algorithms, the simpler implementation of the algorithms as well as the easier data acquisition. [1]

Even though the application of ML-algorithms is well established in other domains, it is not common in the context of production quality. For process optimization in the production quality-domain, physically based modeling (PBM) is commonly used. While PBM offers the advantage of describing the current and future state of a system by physical dependencies, data-driven models use the information from observed data to identify current system characteristics and to predict the future state without requiring a deeper understanding of the physical interdependencies of the process. [4] The development of data-driven models thus shows a high potential for even further optimization of production processes. In the presented case we chose to transform the data into a data-driven model by applying ML-algorithms.

2 Application of Machine Learning in the Production Quality

2.1 Prediction of Product Quality in a Process Chain

In the following, we want to show in a tangible use case at a German manufacturing company that the application of ML-algorithms is worthwhile and to encourage companies to use ML-algorithms for process optimization. To introduce data-driven modeling for process optimization, the Cross-industry standard process for data mining (CRISP-DM) procedure can generally be used [5]. The first step is to understand the corresponding business in more detail. After an initial data acquisition, the characteristics of the data are determined in order to understand the data. The data is subsequently prepared for the application of a suitable ML-algorithm. Based on the data preparation, the implementation of the selected ML-algorithm is described. Finally, the results of the model are evaluated, whereby various criteria are taken into account. Tangible lessons learned will be presented extensively.

The first step of the CRISP-DM is the Business Understanding. The company in this specific use case aims to enhance the efficiency of a process chain, which consists of six different processes. Each product runs through every process sequentially with some processes taking several hours or even days. In order to get a better understanding of the process chain and the corresponding data, we conducted several workshops and web conferences with the company's process engineers. The process chain is depicted in **Fig. 1**.

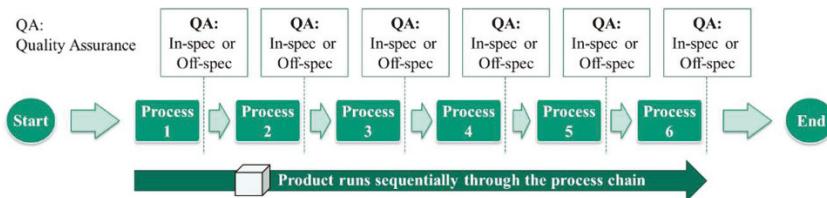


Fig. 1. Illustration of the process chain

Whether a product is an in-spec product can be determined after the completion of each process. Since the cycle time of the entire process chain takes several days, it would be useful to predict whether a product will run out of specification in a process already in earlier stages. If it can accurately predicted that a product will run out of specification, the machines could be equipped with other products. This leads to higher efficiency as well as flexibility of the entire process chain.

Data Understanding as the second step of the CRISP-DM process shows a strong relation to the Business Understanding to the effect that both steps require multiple loops and iterations. The acquired data is stored in separate product-related databases for each of the six processes as semi-structured CSV-files. Due to acquiring a large number of measuring values, there are more than 500 values per process for each product. Different data types like integer, float or string parameters characterize this high

amount of dimensions. Besides a multitude of missing values, the data set is also imbalanced. In this context, an imbalanced dataset means that more products are in-spec than off-spec.

To predict the product quality it is necessary to trace the product data throughout the entire process chain. For that reason, the six different CSV-files need to be linked. This link is created using a product identification number. Since the CSV-files are not uniformly structured, the files need to be transformed multiple times. After the product-related link, the data is cleaned by deleting empty values, apparent correlations as well as by reducing dimensions. Overall, the process of data understanding and preparation took about 80 % of the time regarding the entire CRISP-DM procedure.

In the beginning of the modeling step, a suitable approach how to create a model needs to be selected. Due to the time it takes to learn a data-driven model with an ML-algorithm, only a small number of algorithms can be applied. The process of selecting ML-algorithms depends highly on the use case, the appearance of the data set and the personal experience of the involved data scientists. In this specific case, we interpret the prediction whether a product will be in-spec or off-spec as a classification problem. One class includes all products that run through the process chain being in-spec. Since the quality of the product is measured after each process, the product can become off-spec after each process resulting in six additional classes. Because we are able to label the data set, this multiclass classification problem can be solved using supervised learning algorithms. **Fig. 2** shows a visualization of the processes and the even classes.

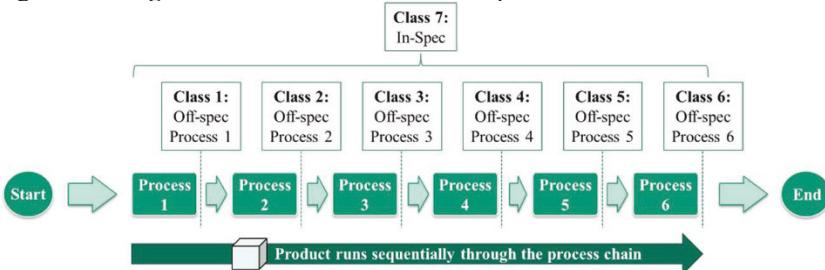


Fig. 2. Visualization of the processes as well as the seven classes

The characteristics of the data set result in the requirements for the algorithm that has to deal with an imbalanced data set, few samples as well as many dimensions. Best practices in other sectors with similar problems are taken from the literature. Besides the results of the literature research, own experiences show beneficial results when decision tree algorithms are applied. Considering the mentioned explanations, the decision tree algorithm Classification and Regression Tree (CART) is selected for this use case [6]. CART can handle high dimensional data sets and has the further advantage that process owners can understand the results of the analysis very quickly and intuitively. The localization, in which the prediction states that the product will run out of tolerance, can be easily detected. Furthermore, the implementation and validation of the decision tree algorithm is simple.

There exist many different platforms for Data Mining as well as ML-algorithm implementation [7]. These platforms can be divided in “Data Science and Machine Learning platforms” like Matlab or RapidMiner and “open source platforms” like Python and

R. Data Science and Machine Learning platforms are characterized by easy handling and fast model development [8]. Nevertheless, operating the platform can result in high licensing costs [9]. Open source platforms like Python and R play an increasingly important role in the data science market because they are free of charge and are the most common programming languages for ML-implementation [8]. We decided to use the “open source platform” Python because the libraries that can be called, such as TensorFlow and scikit-learn, are undergoing strong development. The algorithm is implemented in Python by calling the decision tree algorithm via the scikit-learn library. Scikit-learn uses an optimized version of the CART algorithm [10].

To achieve better performances of the ML-algorithm, hyperparameter must be set. Hyperparameters are the configuration that is external to the model and whose values cannot be estimated from the data set [11]. They are initially set when the algorithm is called by scikit-learn and need to be optimized. Hyperparameters of the decision tree algorithm are e.g. the maximum depth and the minimum size of the tree. There are different approaches to optimize hyperparameters. For this use case, the basic approach, called random search, is applied on the decision tree algorithm. Random search randomly selects any combination of the hyperparameters to be set within an interval of possible hyperparameters. If this combination of hyperparameters lead to better results, the parameters are updated. Basic approaches to set and tune hyperparameters are grid-search and random-search. Over the last years, other tuning approaches like Bayesian Optimization and Gradient Descent became popular [12]. In addition to these advanced approaches, research institutes try to apply heuristics to the hyperparameter tuning problem. These academic approaches include metaheuristics like Particle Swarm Optimization, Ant Colony Optimization and Harmony Search [13].

After running, the performance of the model can be evaluated by a multitude of metrics. The basis of measuring the performance of a classification model is the confusion matrix. The rows of the 2x2 confusion matrix represent the instances in a predicted class while the columns represent the instances in an actual class [14]. If the classification model correctly classifies the input as positive (in-spec) or negative (off-spec), they are considered as true positives (TP) or true negatives (TN). Classifying products falsely as positive or negative counts as false positive (FP) or false negative (FN). Based on the confusion matrix, we can derive different metrics.

Metrics that can be easily derived from the confusion matrix are accuracy and error rate. Other single-value metrics like the F1-Score and Mathew Correlation Coefficient (MCC) are more complex to set up but can still be derived from the confusion matrix. In order to evaluate the performance of the CART algorithm in this specific use case, the MCC is selected. MCC considers imbalanced data sets more efficiently than accuracy and error rate [14]. The mathematical relationship can be taken from equation (1).

$$\text{MCC} = \frac{\text{TP} \cdot \text{TN} - \text{FP} \cdot \text{FN}}{\sqrt{(\text{TP} + \text{FP}) \cdot (\text{TP} + \text{FN}) \cdot (\text{TN} + \text{FP}) \cdot (\text{TN} + \text{FN})}} \quad (1)$$

The MCC considers both mutual accuracies and error rates on both classes. Furthermore, the MCC is a coefficient between the observed and predicted classifications and

returns a value between “−1” and “+1”. A coefficient of “+1” represents a perfect prediction, “0” no better than random prediction and “−1” indicates total disagreement between prediction and observation. [14]

In order to predict the product quality after each process, different CART-algorithms need to be trained because at each process, different amount of data is available to train the CART-algorithm. This leads to four different CART-algorithms, whose performances are depicted in **Fig. 3**. The results include the decision trees that were created after the hyperparameter tuning. By applying random search, the results could be improved by 30% which can be observed in other cases as well [15]. Since no new data is generated in the fourth process, no new decision tree was learned for the change from the fourth to the fifth process.

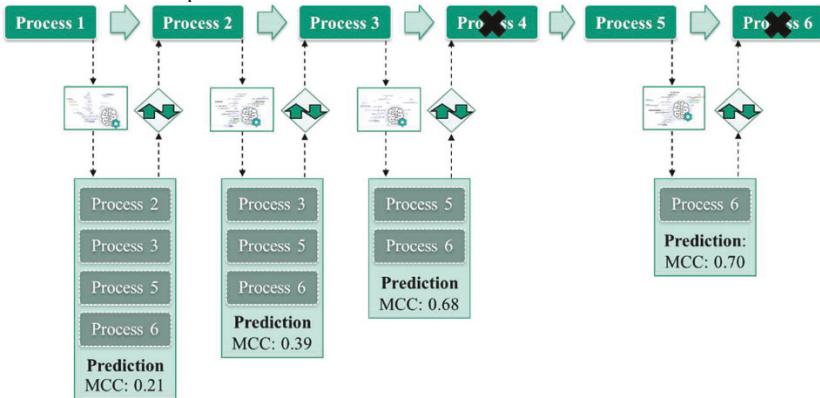


Fig. 3. Performance of the decision tree algorithm

The metric MCC shows the performance of the algorithm in predicting the actual classes of the process. For the first process step the metric is $MCC = 0.21$. This means that there is a match between predicted and actual class, which is relatively low, but better than random prediction. The MCC increases the more processes are accomplished and the fewer processes have to be carried out. The quality of the model improves when more data points are used for the learning task. In addition, less processes and results need to be predicted for the future. After the completion of the fifth process, the metric value is $MCC = 0.70$, which means that the decision tree is a suitable algorithm to predict the product quality sufficiently [16].

2.2 Lessons Learned

In the following, tangible lessons learned are presented, starting with the management level. Then, there will be a focus on the lessons learned for project managers as well as for computer scientists and developers. Two central research needs result from the presented method.

Lessons learned from the managers’ perspective:

- In principle, only available data can be analyzed. Big data only leads to beneficial results if the quality of acquired data is acceptable.

- Managers need to have a precise idea, which results are to be achieved by the analysis of data. Expectations for initial projects must be appropriate.
- Data science projects should start at small scales. As mentioned, simple algorithms like the decision tree can already lead to beneficial results. If the first projects proceed favorable, the projects can be scaled up.

Lessons learned from the project managers' perspective:

- Understanding the business goals and formulating precise objectives is essential when ML-algorithms are implemented.
- A very close collaboration between process engineers and data scientists is unalterable.
- It is natural that many iterations are necessary to get to know technical interdependencies and the characteristics of the data set. The processes of data understanding and data preparation takes a long time compared to the implementation of the ML-algorithm in the end.
- A project manager should be aware of whether it is a quick-win project, low complexity project or long-term commitment.

Lessons learned from the computer scientists and developers' perspective:

- Python notebooks like Jupyter are able to segment the entire code in sensible parts [17]. With Python notebooks, the code can be sequentially updated, which makes coding easier and faster. Introduced variables should still be readable and understandable at the very end of the project.
- Since the selection of a suitable ML-algorithm depends highly on the use case, the appearance of the data set and the personal experience of the involved data scientists, the choice for the ML-algorithm is difficult and sophisticated.
- The hyperparameter tuning is unalterable to solve the classification problem optimally. An optimization based on random search was successful, but advanced optimizations can lead to better solutions.

Overall, we recommend that companies should start with the first data science projects and make their own experiences. Based on first it can be obtained what specific challenges will happen. To describe it in other words - practice makes it perfect!

In addition to the lessons learned, we can derive two central research needs from the presented procedure and the lessons learned. First, it should be evaluated whether more complex hyperparameter optimization methods are capable of outperforming basic approaches like random search and grid search. Second, the procedure of selecting the suitable ML-algorithm was built up on the experiences we had and by comparing the learning task with the literature. A tool supporting us in selecting an appropriate ML-algorithm would have made the process more transparent and reproducible. In the following, we propose a concept how such a tool can function.

3 Selection of Machine Learning-Algorithms

The use of methodologies to solve a specific task creates comprehensible and reproducible results. Therefore, methodologies were developed especially for data mining

and knowledge discovery [18]. Due to the mentioned benefits, they are used in the majority of corresponding projects [3].

CRISP-DM, SEMMA (Sample, Explore, Modify, Model, and Assess) and KDD (Knowledge Discovery in Databases) as the top three methodologies all include a phase specifically designated to create the model for the problem [19]. Due to the generic nature of the three methodologies, the activities in the phase of “Modeling” can be on a different level of complexity ranging from the application of linear regression up to deep learning. Therefore, a data scientist has to decide how to conduct the phase of “Modeling” e.g. by applying an ML-algorithm. Normally the following three aspects are included in this decision: Personal experience, appearance of the data set and literature review. [20]

The problems and corresponding data sets that need to be tackled are domain-specific. Tools that support the data scientist in selecting an ML-algorithm are mostly so called “cheat sheets” [21]. Team members solely bring domain-specific knowledge into the solution. The process of choosing the ML-algorithm is therefore highly dependent on the expertise of the data scientist. Since neither methodologies nor tools include this domain-specific knowledge, the process of selecting the ML-algorithm is not reproducible. Not all domain-specific knowledge can be integrated into a tool. The process of selecting the ML-algorithm stands out by the required creativity of the data scientist. Therefore a decision making tool cannot dismiss the data scientist from his responsibility, but can serve as a support in fulfilling that task. In the following, we present a concept how to set up such a domain-specific decision making tool.

4 Decision Making Tool for Production Quality

The decision making tool (DMT) works as a domain-specific support for the data scientist in selecting an appropriate ML-algorithm to create a model that fulfils problem-specific requirements. This is done by including three main aspects as depicted in **Fig. 4:** Appearance of the data input, requirements of the model to be created and domain-specific knowledge regarding the considered use case. All three factors are included when providing the user a recommendation.

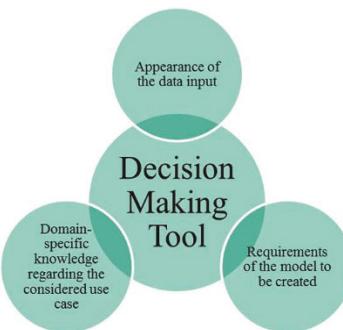


Fig. 4. Factors to be considered when selecting an ML-algorithm

The data scientist interacts with the DMT over a user interface (UI), which he utilizes to describe the specific case he wants to model applying ML-algorithms. The DMT compares the input with historical assessments and problems, including their evaluation. Afterwards the DMT provides the data scientist a list of ML-algorithms probably suitable for the specific use case and additional information about the corresponding selection process. The concept of the DMT is depicted in **Fig. 5** and described in detail in the following.

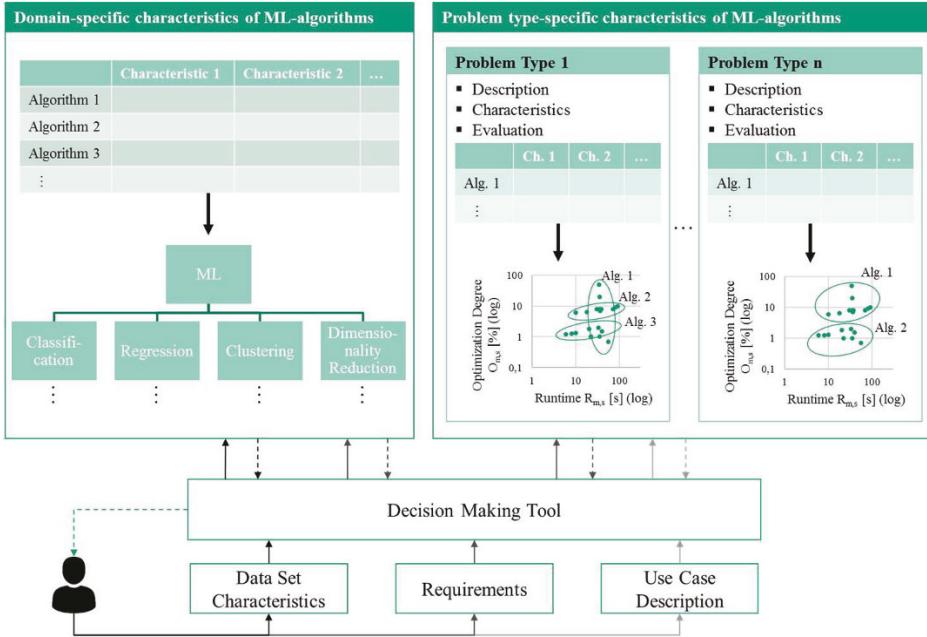


Fig. 5. General Concept of the Decision Making Tool

Using the UI, the data scientist loads the characteristics of the data set, the requirements of the model to be created and a description of the use case into the DMT. Characteristics of the data set are for example the dimensionality of the data, number of features, number of data points, data quality, data distribution or data noise. Requirements of the model to be created are for instance the learning time, performance of the model or transparency of the model. The description of the use case includes information about the type of the use case, e.g. predictive maintenance or product quality prediction. Characteristics like the dimensionality or the maximum running time are quantitative and can directly be loaded into the DMT. Others like the transparency of the model need to be transformed from their qualitative state into a measurable form using for example goal question metrics [22]. This influences the degree of automation to which the characteristics can be loaded into the DMT.

Two main databases function as the backbone of the DMT: A database that includes the domain-specific characteristics of ML-algorithms and a database that stores problem-specific characteristics of ML-algorithms.

The domain-specific characteristics include the attributes of ML-algorithms that are important in the context of the production quality-domain. This includes characteristics and an assessment to which degree the algorithms resp. the learned models meet these characteristics such as interpretability, decomposability, speed, accuracy or learning time. The database is set up and maintained by data scientists working in the production quality-domain.

The problem-specific characteristics are structured by the different types of problems occurring in the production quality-domain such as machine downtime prediction or product failure prediction. For each type of problem, the corresponding description and attributes are available, so that the use case provided by the user can be matched to the most-fitting problem-type in the database. For each problem type from the production quality-domain, different ML-algorithms have been implemented in the past. The information, which algorithms are suitable for the problem-type and the evaluation of their performance is stored accordingly. This is realized by using algorithm maps also known as optimization maps. Each time new types of problems or new evaluations are created, responsible data scientists update the database consequently. This ensures that the specific demands of the production quality-domain and the problem-specific evaluations are considered in the selection process.

The DMT creates a list of algorithms that are promising for the use case by comparing the characteristics of the data set, the requirements of the model to be created and the description of the use case with the historical information stored in the two data bases.

5 Conclusion

In this paper, we presented how ML-algorithms can be applied in a tangible use case from the production quality-domain. In a process chain consisting of six processes, it should be predicted after completion of each individual process whether the product would be off-spec in the following processes. In order to achieve beneficial results, the methodology CRISP-DM was followed. After focusing on the process understanding, data was initially acquired. Afterwards, formats as well as characteristics of the data set has been explored. The preparation of the data comprised the cleaning, transforming and dimensionality reduction in order to apply the ML-algorithm sufficiently. Since we have a multiclass classification problem, the decision tree algorithm CART was selected. The evaluation of the CART algorithm showed that both the methodology and the application of ML-algorithms could lead to beneficial results. On the basis of the mentioned use case, tangible lessons learned could be derived and were divided into lessons learned on the management, project and technology level.

Based on the variety of ML-algorithms, it is difficult to determine, which ML-algorithm is the most suitable for predicting the product quality. In this use case, we compared the performance of different algorithms. These algorithms were selected by the character of the problem, by analyzing the data, by reviewing literature and by the

authors own experience. This process of choosing the ML-algorithm is highly dependent on the expertise of the involved team members. Therefore, a tool that supports the user selecting the ML-algorithm could help in making the process more reliable.

We explained why methodologies are widely used in data mining-projects but why they are just a footnote when choosing ML-algorithm for a specific problem. A concept how a DMT can support data scientists in selecting ML-algorithms for a specific problem was presented. The DMT takes domain-specific demands into account and characterizes ML-algorithms accordingly. Problem type-specific evaluations of ML-algorithms are included in the recommendations. Nevertheless, domain-specific knowledge, expertise regarding selection and implementation of ML-algorithms and the creativity of data scientists will not become obsolete.

6 Funding notes

“The IGF promotion plan 18504N of the Research Community for Quality (FQS), August-Schanz-Str. 21A, 60433 Frankfurt/Main has been funded by the AiF within the programme for sponsorship by Industrial Joint Research (IGF) of the German Federal Ministry of Economic Affairs and Energy based on an enactment of the German Parliament.”

7 References

1. Michael Driscoll (2011) Building data startups: Fast, big, and focused: Low costs and cloud tools are empowering new data startups. <http://radar.oreilly.com/2011/08/building-data-startups.html>. Accessed 14 May 2018
2. Peter Sondergaard (2011) Gartner Says Worldwide Enterprise IT Spending to Reach \$2.7 Trillion in 2012. <https://www.gartner.com/newsroom/id/1824919>. Accessed 14 May 2018
3. Piatetsky-Shapiro G (2014) What main methodology are you using for your analytics, data mining, or data science projects? <https://www.kdnuggets.com/polls/2014/analytics-data-mining-data-science-methodology.html>. Accessed 14 May 2018
4. Datong P. Zhou, Qie Hu, Claire J. Tomlin (2017) Quantitative comparison of data-driven and physics-based models for commercial building HVAC systems
5. Pete Chapman, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer and Rüdiger Wirth CRISP-DM: Step-by-step data mining guide
6. Scikit-learn Developers (2018) Decision Tree Classifier. <http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>. Accessed 14 May 2018
7. Gregory Piatetsky (2018) Survey regarding data mining platforms: What software you used for Analytics, Data Mining, Data Science, Machine Learning projects in the past 12 months? <http://vote.sparklit.com/poll.spark/203792>. Accessed 14 May 2018
8. Gregory Piatetsky (2018) Gainers and Losers in Gartner 2018 Magic Quadrant for Data Science and Machine Learning Platforms. <https://www.kdnuggets.com/2018/02/gartner-2018-mq-data-science-machine-learning-changes.html>. Accessed 14 May 2018
9. Gregory Piatetsky (2017) Forrester vs Gartner on Data Science Platforms and Machine Learning Solutions. <https://www.kdnuggets.com/2017/04/forrester-gartner-data-science-platforms-machine-learning.html>. Accessed 14 May 2018
10. Scikit-learn Developers (2018) Decision Trees. <http://scikit-learn.org/stable/modules/tree.html>. Accessed 14 May 2018
11. Dr. Jason Brownlee (2017) What is The Difference Between a Parameter and a Hyperparameter? <https://machinelearningmastery.com/difference-between-a-parameter-and-a-hyperparameter/>. Accessed 14 May 2018
12. Rafael G. Mantovani, Tomáš Horváth, Ricardo Cerri, Joaquin Vanschoren, André C.P.L.F. de_Carvalho (2016) Hyperparameter Tuning of a Decision Tree Induction Algorithm. IEEE, Piscataway, NJ
13. Paweł Matuszyk, Rene Tatua Castillo, Daniel Kottke A Comparative Study on Hyperparameter Optimization for Recommender Systems

14. Mohamed Bekkar, Dr.Hassiba Kheliouane Djemaa, Dr.Taklit Akrouf Alitouche Evaluation Measures for Models Assessment over Imbalanced Data Sets. 2013
15. Patrick Koch, Brett Wujek, Oleg Golovidov et al. (2017) Automated Hyperparameter Tuning for Effective Machine Learning
16. Thusberg J, Olatubosun A, Vihinen M (2011) Performance of mutation pathogenicity prediction methods on missense variants. *Hum Mutat* 32(4): 358–368. doi: 10.1002/humu.21445
17. (2018) Jupyter Notebook. https://jupyter.readthedocs.io/en/latest/architecture/how_jupyter_ipython_work.html. Accessed 14 May 2018
18. Mariscal G, Marbán Ó, Fernández C (2010) A survey of data mining and knowledge discovery process models and methodologies. *The Knowledge Engineering Review* 25(02): 137–166. doi: 10.1017/S0269888910000032
19. Azevedo A, Santos MF (2008) KDD, SEMMA and CRISP-DM: a parallel overview July 24-26, 2008. Proceedings. In: Abraham A (ed) IADIS European Conference on Data Mining 2008, Amsterdam, The Netherlands, July 24-26, 2008. Proceedings. IADIS, pp 182–185
20. Piatetsky-Shapiro G (2017) Top Data Science and Machine Learning Methods Used in 2017. <https://www.kdnuggets.com/2017/12/top-data-science-machine-learning-methods.html>. Accessed 14 May 2018
21. pakalra, olprod, OpenLocalizationService (2017) Machine Learning – Cheat Sheet für Algorithmen für Microsoft Azure Machine Learning Studio. <https://docs.microsoft.com/de-de/azure/machine-learning/studio/algorithm-cheat-sheet>. Accessed 14 May 2018
22. Basili VR, Caldiera G, Rombach HD (1994) The Goal Question Metric Approach. In: Encyclopedia of Software Engineering. Wiley
23. Pitzer E, Affenzeller M (2012) A Comprehensive Survey on Fitness Landscape Analysis. In: Fodor J, Klempous R, Suárez Araujo CP (eds) Recent Advances in Intelligent Engineering Systems, vol 378. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 161–191

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Which deep artificial neural network architecture to use for anomaly detection in Mobile Robots kinematic data?

Oliver Rettig¹, Silvan Müller¹, Marcus Strand¹, Darko Katic²

¹ Baden-Wuerttemberg Cooperative State University, Department for Computer Science, D-76133 Karlsruhe, Germany,
oliver.rettig@dhbw-karlsruhe.de

² ArtiMinds Robotics GmbH, D-76139 Karlsruhe, Germany

Abstract. Small humps on the floor go beyond the detectable scope of laser scanners and are therefore not integrated into SLAM based maps of mobile robots. However, even such small irregularities can have a tremendous effect on the robot's stability and the path quality. As a basis to develop anomaly detection algorithms, kinematics data is collected exemplarily for an overrun of a cable channel and a bulb plate. A recurrent neuronal network (RNN), based on the autoencoder principle, could be trained successfully with this data. The described RNN architecture looks promising to be used for realtime anomaly detection and also to quantify path quality.

Keywords: neural networks, DL4J, anomaly detection, inertial sensor data, mobile robotics, deep learning

1 Introduction

The navigation of mobile robots typically relies on laser scanner data. Small humps on the floor, e.g. cable channels, doorsills, floor unevenness or other environmental anomalies go beyond its detectable scope. Typically only a 2D map of the environment e.g. 10cm over ground can be established. However, even such small irregularities can have a tremendous effect on the robot's stability and the path quality. Induced vibrations can impact cargo or can reduce the storage life of the robot or its mechanical components.

The new idea of our project is to seek to integrate the detection of small anomalies into dynamic adaptation during the execution of a path and into path planning itself. This should be done based on acceleration data, which can be collected simple and inexpensive by inertial sensors.

Commercial mobile platforms like the Mir-100 allow the definition of driving routes by defining manually a few target points in the map. Then, subsequent path planning is done automatically considering several boundary conditions, e.g. distances to walls. Such a map based path planning can be extended by dynamic path planning in order to adjust to temporary changes in the environment [1]. By driving around or stopping in front of unpredicted and potentially dynamic obstacles collisions can be avoided.

2 Methodology

In robotics typically high-dimensional sensory data with application specific configurations are in use. To make an anomaly detection component reusable without expensive adaptions from specialists, it is desirable to base on a flexible architecture (one or many input channels) and not to use much domain knowledge about the data. This and the need to work with streaming data to find anomalous subsequences instead only single outliers, quantifiable by a score, exclude many anomaly detection methods available in the literature.

On the other side, artificial neuronal networks in general have been used to solve a large range of problems in the field of robotics processing [2] particularly, deep-learning networks are identified as the leading breakthrough technique in the field of mobile robots [3]. They might be used to overcome important challenges in perception and control of mobile robots. For example in [5, 6] a novelty detection in visual data to analyze the robot's environment is described.

In [13] we have shown that a specific deep neural network (DNN) based autoencoder allow for a robust and easily expandable implementation of anomaly detection in kinematic data but which architecture should we use?

There are several approaches. A common way is to train a neuronal network with non anomalous data to be able to predict the next few time frames in the timeseries, based on the current and past values. Then the test data can be compared with the predicted data and the prediction error gives an indication of anomaly [4].

A further class of unsupervised methods combines recurrent neural networks with an encoder/ decoder used as a reconstruction model, where some form of reconstruction error is used, as a score measure of anomaly. The so called autoencoders are trained to reconstruct the normal time-series and it is assumed, that such a model would do badly to reconstruct anomalies, having not seen during training [4].

A newer variant of the autoencoder architecture is the variational autoencoder (VAE) introduced in [7, 8] and amongst others used for anomaly detection [9]. It is based on a reconstruction probability instead a reconstruction error, which should be a more objective anomaly measure. To take into account the temporal structure of timeseries in such an architecture, an additional LSTM [11] layer can be preceded.

3 Concept

The bigger aim of the project behind this paper is to make the usage of mobile robots more robust and flexible by dynamic adaptions to a changing environment. This paper extends the work in [13], which describes in detail the kinematics of the commercially available mobile platform Mir-100 during overrun of a cable channel as a model for an environmental anomaly. Takeoffs are happening particular strong for the rear wheels as a product of the front and the drive wheels already past the cable channel and therefore pulling is more



Fig. 1. Commercially available MiR-100 mobile platform.

effectively. To avoid a damage of the platform or its cargo the idea is to detect the overrun of the front wheels as an anomaly in realtime and to slow down the mobile platform before the rear wheels reach the cable channel.

The measurements described in [13] are done with high precision by a marker based optical system to have a "gold standard". This dataset is also used to train the DNNs presented in this paper.

4 Experiments

Two DNNs are implemented based on DL4J, an open sourced, industry-focused, commercially supported distributed deep-learning framework, which supports multiple CPUs and GPUs.

Furthermore architectures based on a convolutional layer to extract features along the time axis and fed them into a recurrent or dense layer are tried.

The first tested architecture consists of a sequence of four network layers, three of type LSTM [12] with 64, 256 and 100 nodes and hyperbolic tangent as activation function, followed by a dense layer with 100 nodes and linear activation. For fitting the weights, mean squared error is chosen as loss function and RMSPROP, which keeps a moving average of the squared gradient for each weight, as optimizer.

The second architecture consists of six network layers. The first of type LSTM [12] with one input node and 100 output nodes, followed by an variational autoencoder (VAE) introduced in [7, 8] and amongst others used for anomaly detection [9]. It has two encoder- and two decoder-layers, 256 nodes each. The end of the sequence builds a dense output layer.

Both DNNs are trained with vertical acceleration data from the reference dataset which was collected in high precision by a marker based optical system

during driving a mobile platform Mir-100 (Fig. 1) in a gait- and motion analysis lab. Details of the dataset and its acquisition is described in [13]. Three trials are arbitrary chosen to build a validation set.

The DNNs are trained with the remaining 24 example trials with about 15000 time frames each. Only the sections of the trial without the overruns of the cable channel are included in the training set. Over each trial a time window of width 100 frames is moved step by step and the resulting 100 * trial length sequences are mixed up to build the training sequence. To normalize the data and make it more suitable as input for the DNN the mean is subtracted and a division by the standard deviation is done.



Fig. 2. Inertial measurement unit MPU 9250 + Onion Omega2.

Further three test trials with acceleration data (sampling rate 120Hz) are collected from an inertial measurement unit MPU 9250 (Inven Sense) connected via I2C to a Omega2 module (Onion, Fig. 2) and mounted on the mobile platform. To test the DNNs the data is saved in csv files. In principle the data can be streamed via WiFi to an external laptop, which also collects the position data of the mobile platform via the MiRs REST-API.

Vertical acceleration data is collected for three test trials during driving the robot in a corridor with full speed. A cable channel (Fig. 3) is overrun in the middle of the trial.

5 Results

Training of LSTM based autoencoder and the VAE (4) both converges well with a batch size of 50 and a learning rate of 0.2. Loss function values after training with 1 and after 5 epochs are 4.686 and 1.154 for the LSTM layers based autoencoder and 0.619 and 0.039 for the VAE. The values show no differences between the three test trials (optical marker based measurements) for the shown digits.

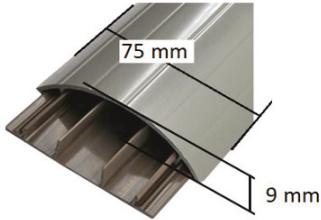


Fig. 3. A cable channel as an anomaly model.

Reconstructed non anomalous data looks very similar in both cases and the overruns of the cable channels are detected clearly as anomaly in all (validation-an inertial sensor based test trials) cases. Fig. 5 shows the difference between original and the predicted/reconstructed data for non anomalous data. The data was normalized to one for the complete trial inclusive anomalous data. That is why the values for non anomalous data in Fig. 5 are so small. Fig. 6 shows a part of the same trial with anomalous data. The three peaks correspond with the overrun of the front-, drive- and rear-wheels. The detections work fine too for inertial sensor based test trials although the DNNs are trained with the marker based optical high precision lab data only.

The approach with a convolutional layer based architecture has no success until now.

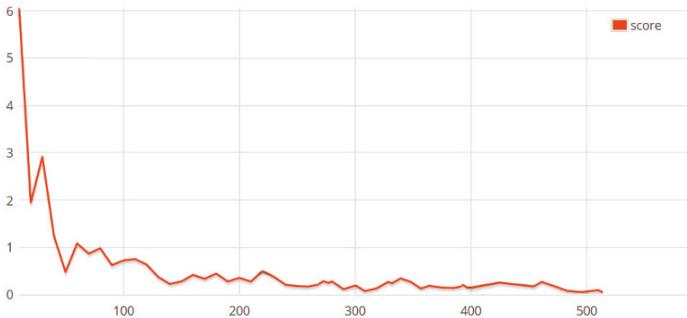


Fig. 4. Score (value of the loss function) over the current minibatch (x-axis), during training of the VAE.

6 Discussion

Anomaly detection works fine for both tested DNN architectures but training of the VAE converges faster and to smaller loss function values which can be an advantage.

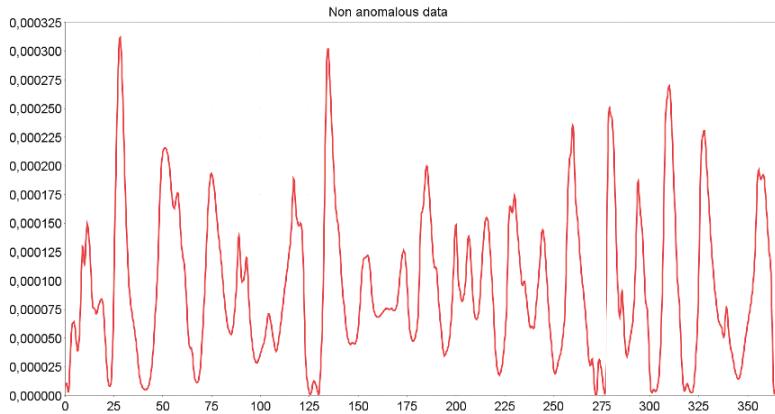


Fig. 5. Normalized anomaly score (predicted minus original acceleration in z-direction) of the VAE based autoencoder; non anomalous data.

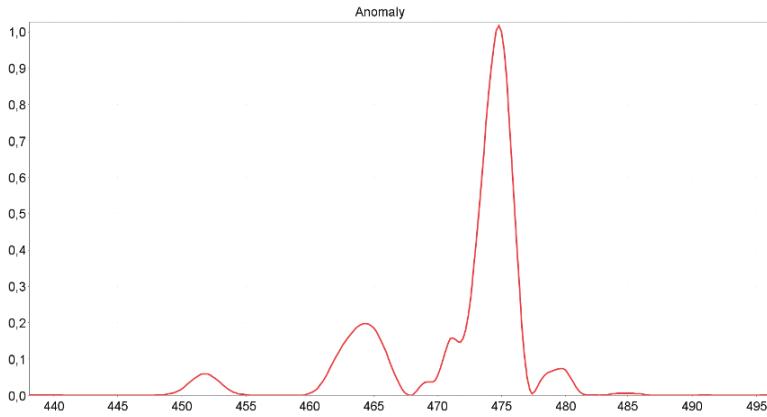


Fig. 6. Normalized anomaly score (predicted minus original acceleration in z-direction) of the VAE based autoencoder; overrun of a cable channel. The three peaks correspond with the overrun of the front-, drive- and rear-wheels. The peak corresponding to the rear-wheels is the biggest one.

These positive results should not hide the fact that a neural net application often needs more care and expenditure in its configuration than an explicit formulated algorithm. Neural nets always come along with the risk to learn hidden but unwanted rules by so called overfitting. In practice you can meet this by a number of arrangements. Carefully chosen architecture details, e.g. for the variational autoencoder used for this project the count of hidden nodes is set higher than the count of input/output nodes. This helps a lot against overfitting. Furthermore you can use so called data augmentation techniques, if the training data set is not diverse enough or too small. To be sure that the DNN learns the concrete paths of the training data as normal, we cut the complete movement paths into pieces and create the training set with a random sequence of these pieces.

If the configuration is such sensitive, why to use a neural net et all? The overrun of the cable channel produces a time window with spikes. With a simple threshold spike detector anomaly detection could be achieved with less effort. Furthermore, this could have the additional advantage that the time threshold for spiky data considered as anomalous, can be defined explicitly, so that the concrete mobile platform is meaningful affected. If only 1D acceleration data is available this can be the better approach.

However, if multichannel data is available e.g. from multiple 3d-acceleration and other sensors in combination and if the algorithm should be robust against single sensor dropouts, the DNN approach is more flexible. It is much easier to train a DNN with a different sensor configuration than to adjust thresholds for multiple sensors and to implement a configuration specific logic to make the system robust against dropouts.

The failure of our convolutional layer approach seems to be caused by a too small training data set.

7 Conclusion and Future Work

The DL4J and its VAE implementation has proved in our project as a production ready framework for anomaly detection in mobile platforms acceleration data. This motivates to implement the newer so called variational recurrent autoencoder (VRAE) [10] based on DL4J. The VRAE extends the VAE and takes into account the dynamic temporal behaviour from the scratch.

The next step is to establish a multichannel approach with three or more 3D acceleration sensors and an optimization of the hyper parameters. For this purpose the DL4J provides the the promising so called Arbiter API.

References

1. Meyer J., Filliat D.: Map-based navigation in mobile robots: II A review of map-learning and path-planning strategies. *Cognitive Systems Research* 4 283-317 (2003)
2. Gamboa, J. C. B.: Deep Learning for Time-Series Analysis. arXiv preprint arXiv:1701.01887. (2017)

3. Tai L., Liu M.: Deep-learning in mobile robotics-from perception to control systems: A survey on why and why not. arXiv:1612.07139. (2016)
4. Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., Shroff, G.: LSTM-based encoder-decoder for multi-sensor anomaly detection. arXiv preprint arXiv:1607.00148. (2016)
5. Neto, H. V., Nehmzow, U.: Real-time automated visual inspection using mobile robots. Journal of Intelligent and Robotic Systems, 49(3), 293-307 (2007)
6. Sofman, B., Neuman, B., Stentz, A., Bagnell, J. A.: Anytime online novelty and change detection for mobile robots. Journal of Field Robotics, 28(4), 589-618 (2011)
7. Kingma, D. P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114. (2013)
8. Rezende, D. J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. arXiv preprint arXiv:1401.4082. (2014)
9. Sölch, M., Bayer, J., Ludersdorfer, M., van der Smagt, P.: Variational inference for on-line anomaly detection in high-dimensional timeseries. arXiv preprint arXiv:1602.07109. (2016)
10. Fabius, O., van Amersfoort, J. R.: Variational recurrent auto-encoders. arXiv preprint arXiv:1412.6581 (2014)
11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation, 9(8), 1735-1780 (1997)
12. Graves, A.: Supervised sequence labelling with recurrent neural networks. Studies in Computational Intelligence 385 (2012)
13. Rettig, O., Müller, S., Strand, M., Katic, D.: Unsupervised Hump Detection for Mobile Robots Based on Kinematic Measurements and Deep-Learning Based Autoencoder. IAS-15 (<http://www.ias-15.org>) 2018 (submitted and accepted)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





GPU GEMM-Kernel Autotuning for scalable machine learners

Johannes Sailer, Christian Frey and Christian Kühnert

Fraunhofer Institute of Optronics, System Technologies and Image Exploitation
IOSB, Karlsruhe, Germany

Abstract. Deep learning (DL) is one of the key technologies in the artificial intelligence (AI) domain. Deep learning neural networks (DLNN) profit a lot from the overall exponential data growth while on the other hand the computational effort for training and inference strongly increase. Most of the computational time in DLNN is consumed by the convolution step, which is based on a general matrix multiplication (GEMM). In order to accelerate the computational time for DLNN different highly optimized GEMM implementations for Graphic Processing Units (GPUs) have been presented in the last years [1] most of these approaches are GPU hardware specific implementations of the GEMM software kernel and do not incorporate the performance dependency of the training data layout. In order to achieve a maximum performance the parameters of the GEMM algorithm have to be tuned for the different GPU hardware and specific data layout of the training task. In this paper we present a two step autotuning approach for GPU based GEMM algorithms. In the first step the kernel parameter search space is pruned by several performance criteria and afterwards further processed by a modified Simulated Annealing in order to find the best kernel parameter combinations with respect to the GPU hardware and the task specific data layout. Our results were carried out on 160 different input problems with the proposed approach an average speedup against the state of the art implementation from NVIDIA (cuBLAS) from around 12 on a NVIDIA GTX 1080 Ti accelerator card can be achieved.

Keywords: GPU, Matrix Multiplication, Autotuning, automatic generation, acceleration, CUDA, BLAS

1 Introduction

1.1 Motivation

Deep learning (DL) is one of the key technologies in the artificial intelligence (AI) domain. Deep learning neural networks (DLNN) profit a lot from the overall exponential data growth while on the other hand the computational effort for training and inference strongly increase. Machine learning applications profit a lot from that overall data growth, since the models can be trained more precise.

However, those algorithms runtime depend heavily on the input data. Most of the computational time in DLNN is consumed by the convolution step, which is based on a general matrix multiplication (GEMM). In order to accelerate the computational time for DLNN different highly optimized GEMM implementations for Graphic Processing Units (GPUs) have been presented in the last years [1]. In order to achieve a high computational throughput, most of these approaches are based on a hardware specific software kernel implementation of the GEMM algorithm. Usually the different hardware dependent kernel parameters are tuned manually, which involves expertise about the specific GPU architecture. Furthermore the performance of the GEMM kernel is strongly affected by the shape of the input data processed different data sizes have a huge impact on the computational runtime of the GEMM kernel due to the different memory layouts of the GPU accelerators.

In order to achieve a maximum performance the parameters of the GEMM algorithm have to be tuned hardware and task specific. In the last years, several autotuning approaches of GEMM kernel parameters have been proposed [2] - the basic idea is to automatically tune a limited number of essential GPU kernel parameters in order to achieve a maximum performance. Usually the approaches do not take into account the size and shape of the given input data, which yields to varying computational runtimes.

The motivation of the presented work is to develop an autotune procedure for GPU based GEMM kernels, which takes into account a comprehensive set of kernel parameters and varying shapes of the data in the input task.

Proposed autotuning solutions such as [2] usually require a lot of computational runtime to find an optimal kernel parameter set. The kernel parameter space e.g. in the MAGMA GEMM kernel [4] is very large and therefore restrictions are made to reduce the search space for the kernel parameters followed by a brute search mechanism. This usually results in high search times for the kernel parameters to be set.

1.2 Related Work

Well known autotuning concepts like the Automated Tuned Linear Algebra Software Project (ATLAS) [5] or the Optimized Sparse Kernel Interface (OSKI) [6] focus on the optimization of CPU calculations. There are only a few approaches, which introduce concepts for autotuning GPU kernel parameters [7] the approaches focus only on a small number of tuning parameters and therefore the achieved performance cannot be compared reasonable to the proposed approach in this work. In order to achieve optimal performance a comprehensive set of GPU kernel parameters have to be taken into account.

In literature there are several more autotuning approaches such as [8,9] . While the work presented in [8] focuses on 3D TFT, the approach in [9] focuses on sparse matrices and optimizing the GPU kernel based on a statistical model. The concepts presented in [10] and [11] focus on automatic generating GPU kernel code and autotune over different generated kernels. Since the generated code is not optimized with respect to the underlying GPU architecture, usually

the performance of these concepts is not optimal. The presented work in this contribution is based on the well-known MAGMA GEMM kernel. The software implementation is characterized by an extensive GPU kernel parameter space. The MAGMA GEMM Kernel, has already been investigated in several autotuning approaches [2,12–16]. The original kernel implementation has been described in [12] and a first autotune concept [13]. With the introduction of the NVIDIA Fermi GPU architecture, the kernel implementation has been revised [14] and an autotuning procedure has been presented in [2]. The approach is characterized by a huge search space for the GPU kernel parameters in conjunction with a brute-force parameter search mechanism, which leads to a high computational effort for finding optimal kernel parameters. With respect to small GEMM operations in [15,16] approaches for batched GEMM operations have been presented and [17] describes the utilization of the Magma GEMM kernel in machine learning procedures. The autotuning approach presented in [18] focuses on energy efficiency of the GPU while processing GEMM operations.

Most of the presented state-of-the-art work is based on a brute-force approach for determining the optimal GEMM kernel parameters. This usually yields to a huge parameter search space and therefore most of the approaches use a parameter combination pre-elimination step in order to reduce the computational effort. The different heuristics for reducing the search space can possibly dismiss optimal kernel parameter combinations. With respect to this suppositions, the presented work focuses on defining optimal heuristics to reduce the search space in combination with a Simulated Annealing(SA) procedure to find efficiently optimal performing GEMM kernel parameters.

2 Solution

Optimal GPU kernel parameters strongly rely on the underlying GPU hardware architecture, the memory layout and the input data size different settings lead to different optimal parameter combinations. Therefore the resulting search space for finding the optimal parameter combination can be enormous. Tuning the parameters by hand is impractical, since it has to be redone for every GPU architecture and every set of input data size again. With respect to these suppositions in the following sections we present a two step autotuning approach for GPU-based GEMM algorithms. In the first step the kernel parameter search space is pruned by several heuristic performance criteria, keeping good performing parameter combinations for a set of different use cases. In the second step based on a modified Simulated Annealing (SA) algorithm the remaining parameter sets are further processed in order to find the best kernel parameter combinations with respect to the GPU architecture and task specific data layout.

In the following sections, the proposed autotuning approach is presented in section 2.1 a short overview of the MAGMA GEMM kernel is given, in section 2.2 we explain the developed heuristics for reducing the search space and in section 2.3 the SA approach is introduced.

2.1 Magma GEMM Stencil structure

The developed autotuning approach is based on the well-known MAGMA GEMM kernel. The original kernel implementation has been described in [12] and is characterized by an extensive GPU kernel parameter space. Algorithm 1 shows the pseudo-code of the kernel. The kernel has 11 parameters - two of the kernel parameters are only relevant for calculations in complex number space. Therefore the kernel parameter space is reduced to nine relevant kernel parameters - the parameters are described in the following:

Blocksizes The Blocksizes BLK_M, BLK_N and BLK_K define how many elements a Threadblock will calculate.

Threadblock dimensions The Threadblock dimensions DIM_X and DIM_Y determine the size of the Threadblock, which calculates a block on the result matrix.

Subdimensions The Subdimensions DIM_XA, DIM_XB, DIM_YA and DIM_YB determine how the Shared Memory(SMEM) is filled.

Algorithm 1: GEMM Kernel Algorithm (simplified)

```

Data: Matrix A [M x K], Matrix B [K x N], Matrix C [M x N], alpha,
       beta
Result: C=A x B * alpha C + beta * B
load  $A_t$  and  $B_t$  to SMEM;
for  $i \leftarrow 0$  to  $K$  step  $BLK\_K$  do
     $A_{t+1}$  and  $B_{t+1}$  to regs;
    for  $i \leftarrow 0$  to  $BLK\_K$  do
        load  $A_t$  and  $B_t$  to REG;
         $C_{temp} = A_t * B_t$ 
    load  $A_{t+1}$  and  $B_{t+1}$  to SMEM;
     $C = C_{temp} * \text{alpha} + \text{beta}$ 

```

2.2 Reducing search space

To reduce the search time for finding optimal kernel parameter sets in the first step it is necessary to eliminate parameter sets, which with respect to the underlying GPU hardware layout are not possible and possibly lead to an unstable behaviour of the kernel execution. The following parameters are reduced:

prelimitations

We started with reducing the viable threadcounts respectively the threadblock dimensions. The threadblock dimensions(DIM_X, DIM_Y) can only be 8, 16 or 32 resulting in 64, 256, 512 or 1024 threads. The GPU manufacturer NVIDIA recommends using a minimum of 64 threads [20], which is the lower limit we are applying, the upper limit is given by the hardware specification of the GPU. Other configurations will not map onto the GPU hardware.

utilization criteria

The idea behind this approach is to make use of the Latency Hiding Principle of the GPU explained in [21]. Basically when the GPU chip loads data from the off-chip Global Memory (GMEM), it will pause the corresponding warp, which is a bundle of 32 threads. The GPU will schedule another warp, while previous one is waiting. Typically loading data from GMEM takes many hundred GPU cycles so Latency Hiding this is essential for performance. To enable Latency Hiding it is essential GPU kernels keep enough warps available and the GPU can switch between contexts while loading data.

The number of available warps on the GPU is described by the utilization. The utilization is limited by the available SMEM and number of Registers (REG) used by the GPU kernel itself. Based on these resources the upper limit of the achievable utilization can be calculated. The resource consumption and the maximum utilization can be determined by analysing the kernel source code - a similar approach can be found in [2]. Important to note is, that the presented work measures the utilization in Warps per Streaming Multiprocessor (SM). The GPU schedules everything in Warps so this seems to be a reasonable approach. Furthermore we are forcing similar utilization levels of SMEM and REG. This constraint avoids parameter combinations, which heavily utilize one resource while barely utilizing the other one. Parameter combinations, which are heavily limited in utilization due to REG suffer from poor performance as well as those, which are heavily limited through SMEM. Those parameter combinations, which are heavily limited in utilization due to SMEM, are keeping to few entries from the result matrix, for the utilization they achieve. Therefore, data has to be loaded more frequently from GMEM than necessary. Parameter combinations, which are highly restricted with REG, are keeping to less data to read for achieving faster times. Therefore, they have to load and wait more frequently.

efficiency criteria

The presented work introduces a further criteria for finding optimal kernel parameters: The efficiency criteria describes how long a parameter combination can work, until data has to be reloaded from GMEM. The efficiency criteria is calculated based on the kernel source code by the equations given in 1 to 3.

- Equation 1 describes how often data is loaded from SMEM, minus how often data is loaded from GMEM.
- Equation 2 describes how often data is read from SMEM compared to loading data from GMEM.
- Equation 3 describes the size of workload per thread.

Equation 1 and 2 prefer combination with high SMEM consumption. Equation 3 prefers squared fields, which are not proven to be better.

$$\begin{aligned} \text{SMEM Accessdiferenz (SMRW)} = & \text{BLK_K} * \\ & ((\text{BLK_M} / \text{DIM_X}) + (\text{BLK_N} / \text{DIM_Y})) - \\ & \text{BLK_K} / \text{DIM_YA} * \text{BLK_M} / \text{DIM_XA} - \\ & \text{BLK_N} / \text{DIM_YB} * \text{BLK_K} / \text{DIM_XB} \end{aligned} \quad (1)$$

$$\begin{aligned} \text{SMEM Reuse (SMR)} = & \text{BLK_K} * \\ & ((\text{BLK_M} / \text{DIM_X}) + (\text{BLK_N} / \text{DIM_Y})) / \\ & (\text{BLK_K} / \text{DIM_YA} * \text{BLK_M} / \text{DIM_XA} + \\ & \text{BLK_N} / \text{DIM_YB} * \text{BLK_K} / \text{DIM_XB}) \end{aligned} \quad (2)$$

$$\text{Work per Thread (WpT)} = (\text{BLK_M} / \text{DIM_X}) * (\text{BLK_N} / \text{DIM_Y}) \quad (3)$$

Because of the contradictory definition of the efficiency criteria and the utilization criteria, it is not possible to optimize both at once. The efficiency criteria will force contexts, which will reduce the reload operations from GMEM and therefore enforce higher resource consumption. On the other hand, the utilization criteria will favour shorter working times for the contexts by consuming less SMEM and REG resources. The approach of this work is to use those parametrizations for the subsequent SA autotuning step, which forces to achieve the highest efficiency criteria on a specific utilization level. This ensures long living contexts on a specific utilization level with respect to the latency hiding principle from Paragraph 2.2. With respect to these suppositions, the resulting search space reduces to 84 meaningful parameter combinations.

2.3 Simulated Annealing

Simulated annealing (SA) is a probabilistic technique for finding optimal parameter combinations in a given search space - a detailed overview of the concept is given in [22]. For our approach SA is fitting, because of its ability to ignore local minima and converge to the global one. Sorting the search space after different criteria enforces grouping of parameter combinations with similar runtime on similar problems in the search space, resulting in faster convergence of SA. The parameter combinations found in Paragraph 2.1 are sorted according to their achieved utilization on the GPU and processed in the SA step. It should be noted, that other possible criteria for SA could be the blocksizes ($\text{BLK_M} * \text{BLK_N}$) or the leading dimension (DIM_X) from Paragraph 2.1.

3 Performance Evaluation

The performance evaluation of the proposed work is based on a NVIDIA Pascal GPU (MSI Geforce GTX 1080 Ti Aero 11G OC) in combination with a Intel

Xeon E5-1620 with 96 GB Memory host system. The operating system is Windows 64 Bit with NVIDIA Driver Version is 390.65 and CUDA 8. To evaluate the performance of the proposed approach different data sets are used - Table 1 gives an overview of the different matrix shapes for evaluation. These matrix shapes have been chosen, because cuBLAS proven to perform very well. An evaluation test consists of three Matrices A,B and C with format M x K, K x N and M x N \in N. Additionally in order to illustrate the flexibility of the proposed approach, several other matrix shapes have been evaluated. The results of the performance evaluation are shown in Figure 1 and Table 2. Figure 1 shows the achieved speedups with respect to the matrix shapes compared to cuBLAS. It can be seen, that the larger N the lower the performance speedup. In the worst case the achieved result of the proposed approach is 1.3 times faster than the highly optimized cuBLAS routine, in the best case the speedup is 187 times faster than cuBLAS.

Table 1 shows a comparison between the best-found solutions with a standard the brute-force approach to the proposed approach based on SA proposed in this work. The speedup for finding optimal kernel parameters with the proposed SA approach is nearly five to six times faster than the standard brute force approach, while the performance loss for GEMM kernel execution is maximum 10%.

Algorithm 2: Procedure for proving performance capability of this work. The algorithm generates examples in the form of three matrices A, B and C with the formats M x K, K x N and M x N \in N. After 152 generated examples the process terminates.

```

for  $M = 25; M < 1000000; M = M + 25$  do
  for  $K = 25; K < 1000000; K = K + 25$  do
    if  $M * K = 6250000$  or  $25000000$  or between 2000 and 1000
      then
        N = 25;
        Brute-force search space (M,N,K);
        Simulated Annealing (M,N,K);
        N = 0.5 * M;
        Brute-force search space (M,N,K);
        Simulated Annealing (M,N,K);
        N = M;
        Brute-force search space (M,N,K);
        Simulated Annealing (M,N,K);
        N = 5 * M;
        Brute-force search space (M,N,K);
        Simulated Annealing (M,N,K);

```

Matrix Entries			Matrix Number	Matrix Entries			Matrix Number
M	K	N		M	K	N	
10000	25	10000	1	1250	5000	1250	5
10000	200	10000	2	25	1000000	25	6
5000	500	5000	3	4000	50000	25	7
2500	2500	2500	4	25	10000	2000	8

Table 1. Data Matrix sizes for performance evaluation.

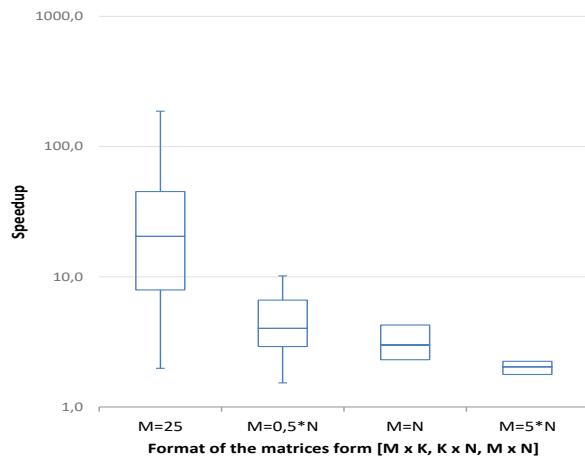


Fig. 1. Comparison of the speedup times against cuBLAS with the brute-force approach on the examples from Algorythm 2. The minimum Speedup was 1,3, the maximum was 187 times as fast as cuBLAS. The average was 12.3 compared to 11.9 in the Simulated Annealing approach. The figure shows, that with an increasing size of N compared to M the speedup reduces. But there was no negative speedup in this test so the results are always faster than the calculation with cuBLAS.

Matrix Format	brute-force
	Speedup against Simulated Annealing (%)
N=25	10,0
N=M/2	8,4
N=M	3,7
N=5M	4,3
average	5,8

Table 2. Comparison between the best achieved brute-force solution in comparison to the found solution with the Simulated Annealer on examples in the form of three matrices A, B and C with the formats $M \times K$, $K \times N$ and $M \times N \in N$

4 Conclusion

The computational throughput of Machine Learning algorithms is limited by the available computational power of the underlying hardware. Most of the computation power in DLNN is consumed by the convolution step, which is based on a general matrix multiplication (GEMM). To accelerate the computational time in Machine Learning applications different highly optimized GEMM implementations for GPUs have been presented in the last years - usually these software libraries have been optimized for a specific GPU version and a specific layout of the data to be processed.

In order to achieve a maximum performance the kernel parameters of the GEMM algorithm have to be tuned hardware and learning task specific. With respect to these suppositions, we have presented a two-step autotuning approach for GPU-based GEMM algorithms: In the first step, the kernel parameter search space is pruned by analysing the kernel source code with several developed performance metrics. In the second step a modified Simulated Annealing algorithm is utilized, which enables a fast searching process for performance optimal kernel parameters, while maintaining search runtimes lower than state of the art brute-force implementations. We have shown that the proposed approach for autotuning MAGMA-GEMM kernels yields high performance and adapts to the GPU hardware and the data layout. Our results have been carried out base on 160 different input problems - we get an average speed up against the state of the art GEMM implementation from NVIDIA (cuBLAS) from around 12 on Pascal based NVIDIA accelerator cards. The key concepts of this contribution can be generalized, to autotune the kernel parameters of other performance sensitive GPU kernels.

5 Acknowledgements

This work was developed in the Fraunhofer Cluster of Excellence "Cognitive Internet Technologies".

References

1. Theano: Deep learning on gpus with python Bergstra, James and Bastien, Frédéric and Breuleux, Olivier and Lamblin, Pascal and Pascanu, Razvan and Delalleau, Olivier and Desjardins, Guillaume and Warde-Farley, David and Goodfellow, Ian and Bergeron, Arnaud and others, NIPS 2011, BigLearning Workshop, Granada, Spain
2. Autotuning GEMMs for Fermi Jakub Kurzak, Stanimire Tomov, Jack Dongarra 2011
3. Fast k nearest neighbour search using GPU Vincent Garcia, Eric Debreuve, Michel Barlaud available at: <http://vincentfp Garcia.github.io/kNN-CUDA/> access 13.06.2018
4. Magma project page <http://icl.cs.utk.edu/magma/> access 13.06.2018

5. Automated empirical optimization of software and the ATLAS project R. Clint Whaley and Antoine Petitet and Jack J. Dongarra 2001
6. OSKI: A library of automatically tuned sparse matrix kernels Richard Vuduc and James W. Demmel and Katherine A. Yelick 2005
7. Application-independent Autotuning for GPUs Martin TILLMANN and Thomas KARCHER and Carsten DACHSBACHER and Walter F. TICHY 2013
8. Auto-Tuning 3-D FFT Library for CUDA GPUs Akira Nukada and Satoshi Matsuoaka 2009
9. Performance Prediction Based on Statistics of Sparse Matrix-Vector Multiplication on GPUs Ruixing Wang and Tongxiang Gu and Ming Li 2017
10. Automatic C-to-CUDA Code Generation for Affine Programs Muthu Manikandan Baskaran and J. Ramanujam and P. Sadayappan 2010
11. A Script-Based Autotuning Compiler System to Generate High-Performance CUDA Code MALIK KHAN and NUST PROTOMU BASU and GABE RUDY and MARY HALL and CHUN CHEN and JACQUELINE CHAME 2013
12. Benchmarking GPUs to Tune Dense Linear Algebra Vasily Volkov, James W. Demmel 2008
13. A Note on Auto-tuning GEMM for GPUs Yinan Li1, Jack Dongarra, Stanimire Tomov 2009
14. An Improved Magma Gemm For Fermi Graphics Processing Units Rajib Nath, Stanimire Tomov, Jack Dongarra 2010
15. Performance, Design, and Autotuning of Batched GEMM for GPUs Ahmad Abdelfattah, Azzam Haidar, Stanimire Tomov and Jack Dongarra 2016
16. Novel HPC Techniques to Batch Execution of Many Variable Size BLAS Computations on GPUs Ahmad Abdelfattah, Azzam Haidar, Stanimire Tomov and Jack Dongarra 2017
17. Brute-Force k-Nearest Neighbors Search on the GPU Shengren Li and Nina Amenta 2015
18. Experiences in autotuning matrix multiplication for energy minimization on GPUs Hartwig Anzt, Blake Haugen1, Jakub Kurzak1 and Piotr Luszczek and Jack Dongarra 2015
19. Model-driven Autotuning of Sparse Matrix-Vector Multiply on GPUs Jee W. Choi, Amik Singh and Richard W. Vuduc 2010
20. <https://www12.informatik.uni-erlangen.de/edu/map/ss08/talks/> Best Practices for GPU Programming.ppt, Access 26.2.2018 16:21. Best Practise for GPU Programming.
21. Understanding Latency Hiding on GPUs, Vasily Volkov, 2016
22. AARTS, Emile; KORST, Jan. Simulated annealing and Boltzmann machines. 1988

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Process Control in a Press Hardening Production Line with Numerous Process Variables and Quality Criteria

Anke Stoll, Norbert Pierschel, Ken Wenzel, and Tino Langer

Fraunhofer Institute for Machine Tools and Forming Technology IWU,
Reichenhainer Str. 88, 09126 Chemnitz, Germany
anke.stoll@iwu.fraunhofer.de

Abstract. Today, the optimization of the press hardening process is still a complex and challenging task. This report describes the combination of linear regression with least squares optimization to adjust the process parameters of this process for quality improvement. The FE simulation program AutoForm was used to model the production line concerned and various process and quality parameters were measured. The proposed system is capable of automatically adjusting the process parameters of following process steps based on the quality estimate at each step of the production line. An additional benefit is the identification of likely defective parts early in the production process. Based on the results derived from 1000 observations a better understanding of the process was obtained and in the future the combined regression and optimization approach can be extended to more complex production lines.

Keywords: linear regression, least squares optimization, production line, press hardening, process control

1 Introduction

One of the goals of Industry 4.0 is the optimization and customization of production processes through digitization with algorithms, big data approaches and high technologies [1]. Currently, machine learning (ML) approaches support monitoring, diagnosis and (off-line) system optimization for fault detection, maintenance, decision support and product quality improvement [2,3]. The field of ML is manifold and various different methods are available. However, in manufacturing and other fields of application the complexity of ML methods can hinder their adoption even though the data acquisition for many production processes is possible and a sufficient data base is available or can be obtained. Therefore, this work aims to implement a simplistic ML and optimization approach for a production line. The paper starts with a discussion of work related to ML and process control in Section 2, followed by the presentation of the methodology in Section 3, that includes a description of the data sets, the data preparation, and the estimation techniques. The results of the analysis are described in Section 4. Section 5 presents the conclusions and discussion of practical implications.

2 State of the Art

First approaches for process control based on ML were conducted by Oh and co-workers [4] who apply Neural Network/Partial Least Squares to model the relationship between multiple process parameters and multiple quality parameters in the production process of metal plates of a complex structure. Senn and co-workers [5] use Principal Component Analysis and Artificial Neural Networks to model the relation between observed quantities and state variables for a deep drawing process. However, comprehensive studies for ML based process control within production lines are still sparse. In order to contribute to fill this gap we propose an intuitive approach to intelligently control the process parameters within a production line for quality improvement of the final product. The introduced intelligent system is based on linear regression and least squares optimization.

3 Data and Methods

We consider a production line for the press hardening of sheet metal in order to produce center pillars, which are ultra-high-strength car body parts. Here, we will focus on the three process steps warming, handling and quenching, see Figure 1. The process involves inserting sheets, which have been heated beyond the austenitizing temperature of about 900°C, into a cooled forming tool, in which they are then quenched. The thermal integrated processing produces press-hardened parts with an extremely high tensile strength of up to 1,500 MPa for the ultra-high-strength steel 22MnB5. The handling of the sheets is done by robots.

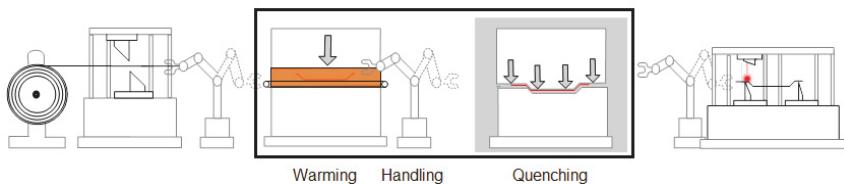


Fig. 1. Production line for the press hardening of sheet metal focusing on the three steps: (1) warming in a furnace unit, (2) handling with a robot system with grippers, and (3) quenching.

Similar to Oh et al. [4] each process can be described by its

- uncontrollable factors (initial conditions of materials or processes; and fix variables),
- controllable factors (adjustable variables) and
- quality variables (response variables representing the final product quality).

Figure 2 shows the parameters we considered in our case study. Uncontrollable input variables are the sheet thickness (ST) and the tool temperature during quenching (ToTemp). Controllable input variables are sheet temperature after warming (STemp), transfer time between warming and quenching (TT), spacing (Sp), quenching time (QT) and quenching force (QF). Quality variables are the output variables hardness at a critical point P1 on the finished part (P1H) and sheet thickness at another critical point P2 (P2ST). The ML method proposed in the next section then correlates input and output variables and allows process intervention for quality improvement. Data were acquired using the sheet metal forming software AutoForm [6], similar to [7]. The whole data set consists of 1000 observations which were achieved by variation of the input parameters as shown in Table 1.

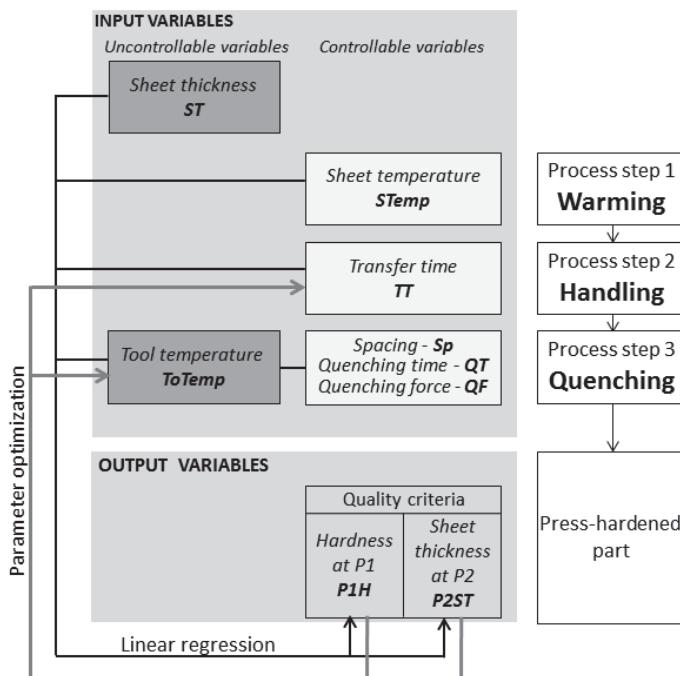


Fig. 2. Production line with three process steps and their respective controllable and uncontrollable variables. Linear regression is conducted based on the existing database. After the warming process is finished, parameter optimization for the process steps handling and quenching is possible.

3.1 Data Preparation

The open source statistical programming tool R [8,9] was used to evaluate the data generated by AutoForm. The aim of this study was to find an appropriate

ML model to describe the relationship between the input and output parameters. Upper and lower boundaries for the allowed input parameter variations are defined as stated in Table 1. Boundaries for the quality criteria have to be defined as well. These depend on the type of component that is produced. The focus can be on maximum component hardness or for example on the maximum thickness of the finished component. As we focus on a part from the automotive industry we want to maximize/increase both, the sheet thickness and hardness at critical points which are prone to tearing. Thus, no upper boundaries for P1H and P2ST were defined.

Table 1. Process parameters, quality criteria, and regression coefficients for the estimation of P1H and P2ST.

Variable	lower	upper	default		
	boundary	boundary		coefficients P1H	coefficients P2ST
ST [mm]	1.45	1.55	1.5	-	9.4×10^{-1}
STemp [°C]	900	950	900	1.1	-
TT [s]	5	55	5	-6.2	4.9×10^{-4}
ToTemp [°C]	80	300	190	-7.4×10^{-2}	-5.3×10^{-6}
QF [kN]	500	2,500	2,000	3.5×10^{-6}	-1.3×10^{-9}
QT [s]	2	65	2	3.3×10^{-1}	-1.2×10^{-4}
Sp [mm]	0.1	2	1.05	1.1	2.4×10^{-3}
Intercept				-4.5×10^2	6.2×10^{-2}
P1H [HV]	390				
P2ST [mm]	1.43				

3.2 Linear Regression for Quality Prognosis

Description of the Model Ultimately, we aim for on-line process control which makes the application of high speed models and fast predictions necessary. As a first step – conducted off-line – we need to describe the relationship between input and output variables in a distinguishable way. A general linear model which accounts for the single parameters linear effects was considered. In general, a linear regression equation has the following form

$$\text{DepVar} = a + (b_1 \times \text{IndepVar}_1) + \dots + (b_n \times \text{IndepVar}_n).$$

Where a, b_1, \dots, b_n are unknown parameters, DepVar stands for dependent variable and represents the qualities P1H and P2ST, respectively. IndepVar's are the independent variables, such as the process parameters.

The analysis is carried out in R using the lm() function for fitting linear models independently for the two quality parameters P1H and P2ST. The resulting regression coefficients are shown in Table 1.

Validation of the Model The regression analysis indicates that STemp, TT, ToTemp, QF, QT and Sp had significant influence on P1H, which is confirmed by the *p*-values (no significant influence of ST). The overall suitability of a linear regression approach is supported by an adjusted R^2 of 0.90 which describes the percentage of the dependent variable variation by the model. P2ST can be thoroughly described by linear combinations of ST, TT, ToTemp, QF, QT and Sp (no significant influence of STemp) with an adjusted R^2 of 0.99.

Since the total number of observations is limited and a partition into training and test data is not sensible without loosing significant modeling capability the models were validated with K -fold cross-validation. For $K = 5$, the overall mean square of prediction error is 97.6 for the linear model (compared to 102 for the complete model with all variables) to predict P1H and 3.87×10^{-6} for the prediction of P2ST (compared to 6.14×10^{-6} for the complete model). This indicates reasonably good linear models despite the limited number of observations which will be increased in the future.

3.3 Least Squares Optimization with Constraints

Set-up of the Optimization Problem After each step in the production line the qualities P1H and P2ST are estimated using the variables already measured in combination with assumptions for variables of the process steps not yet performed (default values in Table 1). These assumptions are based on technological expert knowledge. After the warming process, we know the ST and the STemp. In order to get a first estimate for the expected quality P1H and P2ST we use the linear regression model established in 3.2 with the measured ST and STemp and default values for TT, ToTemp, Sp, QT and QF as stated in Table 2. If the estimated quality is below the predefined threshold, also stated in Table 2, the controllable variables in following process steps have to be adjusted in order to bring the quality back into its desirable interval. An optimization process was established, which calculates the necessary adjustments. Least squares are applied to solve the emerging inhomogeneous linear system with constraints after every process step. With each step the accuracy of the model improves as less and less process estimates have to be used to predict the quality.

In order to solve the optimization problem least squares with equality and inequality constraints is performed. The function from the R-package limSolve is called lsei() and solves

$$\min ||Ax - b|| \text{ subject to } Ex = f, Gx \geq h.$$

For the optimization after process step 1 (warming) the matrix A is the unity matrix with dimension 4 because there are four subsequently determinable variables left in the manufacturing process. The vector b contains the default values / desirable process values for the 4 adjustable variables. The objective function tries to find a solution for the 4 adjustable variables which is as close as possible to the desired default values. Since our optimization problem does not have equalities, E is a zero matrix of the dimension 4 and f is a vector of zeros. The inequality constraint $Gx \geq h$ is constructed from the upper and

lower boundaries of the adjustable variables and the linear regression equations combined with the quality boundaries. The optimization after process step 2 is conducted in a similar way but only 4 adjustable variables are remaining.

Weighting of Parameters Since an adjustment of some parameters is easier than others, e.g. TT or QT, weighted least squares can be used to improve the efficiency of parameter optimization. The weighting vector W_a as an additional input for the lsei() optimization function is defined to prefer changes on easily adjustable variables such as TT and QT. Thus the weighting coefficients for TT and QT were chosen 1 while they are 100 for QF and Sp. By giving each variable its proper amount of influence on the resulting quality a more realistic image of the real press hardening process is established. The weight for each variable is given relative to the weights of the other variables.

4 Results and Discussion

In order to show the versatility of the approach four different scenarios are presented in the following.

The type of component to be produced has an immediate impact on the optimization problem. The system can be optimized towards the hardness of the produced component, process velocity (usually as fast as possible to be cost-effective), geometric accuracy or other objectives. In the production industry the overall equipment effectiveness (OEE) is a relevant and popular indicator for a machine or production line. Thus, we want to focus on a process as fast as possible which correlates directly with the maximization of the number of cycles in a production line. For this purpose, the default setting for TT is the smallest possible value of 5 s, similar to a minimum QT of 2 s. The QF has to be as high as possible in order to allow the quenching process to be fast. Thus, a QF of 2,000 kN is chosen as default allowing slight upward adjustment with a total maximum of 2,500 kN. The Sp default is 1.05 mm.

The quality control of the production line can return an “accepted part” for parts meeting both quality criteria as defined in Table 1 and “defective part” otherwise.

4.1 No Parameter Adjustment Necessary

In the majority of cases a production line should produce high quality parts when working with feasible process parameter intervals. One example for a process cycle resulting in an accepted part is shown in Table 2. The warming process is conducted with a ST of 1.5 mm and a resulting STemp of 900°C. Both P1H and P2ST are estimated with the linear regression approach described in section 3.2 with default values for TT, QF, QT and Sp (see Table 1). The predicted P1H and P2ST imply a qualitatively accepted part. Even with a longer than targeted TT of 10 s instead of 5 s the quality at the end of the process is still within range (Table 2, row 3) and no parameter adjustment is necessary (Table 2, row 4).

Table 2. No adjustment necessary. Highlighted in gray are the process parameters already known.

process step	process 1		process 2		process 3		quality	
	ST	STemp	TT	QF	QT	Sp	P1H	P2ST
	[mm]	[°C]	[s]	[kN]	[s]	[mm]	[HV]	[mm]
process 1 - <i>default</i>	1.5	900	5	2,000	2	1.05	483.8	1.47
process 1 - <i>adjusted</i>	1.5	900	5	2,000	2	1.05	483.8	1.47
process 2 - <i>start</i>	1.5	900	10	2,000	2	1.05	452.3	1.47
process 2 - <i>adjusted</i>	1.5	900	10	2,000	2	1.05	452.3	1.47

4.2 Parameter Adjustment

If the ST is 1.45 mm instead of 1.5 mm with an identical STemp of 900°C the estimated P2ST is too low. If the process is not adjusted, this cycle will likely produce a rejected part. However, the proposed approach allows an adjustment of the parameters in process step 2 and 3 in order to produce an accepted part. The model suggests a TT of 19.8 s instead of 5 s, a maximum QF of 2,500 kN and a slightly increased Sp of 1.2 mm in order to obtain a part with the required sheet thickness (Table 3, row 2). If the suggested TT of 19.8 s is slightly longer with 20.5 s, P1H is outside the feasible interval and an adjustment in process 3 is necessary (Table 3, row 3). Here, the QT is increased to 12.8 s and the Sp is increased to 1.71 mm in order to obtain a part with accepted quality (Table 3, row 4).

Table 3. Parameter adjustment. Written in bold are violated quality criteria.

process step	process 1		process 2		process 3		quality	
	ST	STemp	TT	QF	QT	Sp	P1H	P2ST
	[mm]	[°C]	[s]	[kN]	[s]	[mm]	[HV]	[mm]
process 1 - <i>default</i>	1.45	900	5	2,000	2	1.05	480.8	1.42
process 1 - <i>adjusted</i>	1.45	900	19.8	2,500	2	1.2	390.0	1.43
process 2 - <i>start</i>	1.45	900	20.5	2,500	2	1.2	385.9	1.43
process 2 - <i>adjusted</i>	1.45	900	20.5	2,500	12.8	1.71	390.0	1.43

4.3 Limited Adjustment

For an even lower sheet thickness of 1.44 mm and STemp of 900°C the quality criterion P2ST is violated with 1.41 mm instead of 1.43 mm. An adjustment of the process parameters of process 2 and 3 is not possible without violating some of the constraints as the optimization approach aims for keeping both quality criteria within their intervals and at the same time all process parameters within their boundaries. Thus, a limited adjustment is performed in order to obtain a part as close as possible to accepted quality (Table 4, row 2). The parameters of process 2 and 3 are altered such that P1H is just at the lower limit and P2ST

is improved as much as possible (1.423 mm instead of 1.41 mm and close to accepted quality). For this purpose, TT is increased from 5 s to 19.8 s, QF is at its maximum and Sp is increased to 1.99 mm.

Table 4. Limited adjustment / no adjustment possible. Again, violated quality criteria are written in bold. Marked with a star are improved by still violated qualities.

process step	process 1		process 2		process 3		quality	
	ST	STemp	TT	QF	QT	Sp	P1H	P2ST
	[mm]	[°C]	[s]	[kN]	[s]	[mm]	[HV]	[mm]
process 1 - <i>default</i>	1.44	900	5	2,000	2	1.05	480.2	1.41
process 1 - <i>adjusted</i>	1.44	900	19.8	2,500	2	1.99	390.0	1.423*
process 2 - <i>start</i>	1.45	900	20.5	2,500	2	1.99	385.4	1.423
process 2 - <i>adjusted</i>	1.45	900	20.5	2,500	2	1.99	385.4	1.423

4.4 No Adjustment Possible

Sometimes the quality prognosis after process step 1 indicates that the produced part will not meet the final product quality requirements. Given the fact, that the prognosis is accurate, this is a very valuable information this early on in a production line because defective parts can be removed early in the production process with the additional benefit of cost and energy savings. Table 4 shows an example where after process step 2 no parameter adjustment is possible without violating the constraints. HP1 and P2ST will both be too low no matter how the process parameters in process 3 are altered.

5 Discussion and Conclusions

A combination of linear regression and least squares optimization can be employed to reproduce a bidirectional relation between process parameters and quality parameters in a fast and reliable manner. The proposed system is capable of estimating the quality outcome at any step of a production line. It allows adjustment of the controllable variables one or more process steps further on and identifies defective parts early in the production process.

If more than one quality criterion is considered, conflicting relations between them have to be expected. The goal of the parameter optimization is that parameter adjustments are found such that all quality criteria are satisfied. This constraint may result in an unsolvable optimization problem. The unsolvability of the problem after the first production step (or later in the process) indicates that the final product might not satisfy at least one quality criterion. The quality prognosis this early on in the production process is a valuable information, as potentially defective parts can be sorted out early. This saves resources, machine time and energy.

The accuracy of the quality prognosis is mainly driven by the accuracy of the regression model. Therefore, a sufficiently large database is necessary. In the future, we plan to increase the data volume for a higher prognosis accuracy. The simulated data should be as close as possible to reality. Typically, the parameters of the press-hardening cycles follow a normal distribution and most of these cycles produce accepted parts. However, in the AutoForm software, a mesh is placed over the boundaries of the process parameters and the parameter variations are evenly distributed over the mesh. How this affects the regression model remains to be investigated. A validation of the ML approach with FEM simulations is under way. Once the extended simulation based regression and optimization approach works we will move on to experimental data and other more complex production lines.

Acknowledgments. This work was supported by the Fraunhofer-Gesellschaft with the funding of the lead project “ML4P – Machine Learning 4 Production”. Furthermore we thank the European Union, the Free State of Saxony as well as the Fraunhofer-Gesellschaft for the funding of the High Performance Center Smart Production. Many thanks to Thomas Lieber for acquiring the simulation data.

References

1. Lu, Y.: Industry 4.0: a survey on technologies, applications and open research issues. *Journal of Industrial Information Integration* 6, 1–10 (2017)
2. Harding, J.A., Shahbaz, M., Kusiak, A.: Data mining in manufacturing: a review. *Journal of Manufacturing Science and Engineering* 128.4, 969–976 (2006)
3. Niggemann, O., Stein, B., Maier, A.: Solving Modeling Problems with Machine Learning – A Classification Scheme of Model Learning Approaches for Technical Systems. In *Model-Based Development of Embedded Systems (MBEES)*, Dagstuhl (2012)
4. Oh, S., Han, J., Cho, H.: Intelligent process control system for quality improvement by data mining in the process industry. *Data mining for design and manufacturing*. Springer US, 289–309 (2001)
5. Senn, M., Link, N.: A universal model for hidden state observation in adaptive process controls. *International Journal on Advances in Intelligent Systems* 4(3-4), 245–255 (2012)
6. AutoForm, url: <https://www.autoform.com/>
7. Neugebauer, R., Schieck, F., Polster, S., Mosel, A., Rautenstrauch, A., Schönher, J., Pierschel, N.: Press hardening An innovative and challenging technology. *Archives of civil and mechanical engineering* 12(2), 113–118 (2012)
8. Ihaka, R., Gentleman, R.: R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics* 5(3), 299 – 314 (1996)
9. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, <https://www.R-project.org/> (2017)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





A Process Model for Enhancing Digital Assistance in Knowledge-Based Maintenance

Klaudia Kovacs^{1,2*}, Fazel Ansari^{1,2}, Claudio Geisert³, Eckart Uhlmann^{3,4},
Robert Glawar², Wilfried Sihn^{1,2}

¹Vienna University of Technology (TU Wien), Institute of Management Science

²Fraunhofer Austria, Division of Production & Logistics Management, Vienna, Austria

³Fraunhofer Institute for Production Systems and Design Technology IPK, Berlin, Germany

⁴Institute for Machine Tools and Factory Management, TU Berlin, Berlin, Germany

klaudia.kovacs@tuwien.ac.at

Abstract. Digital transformation and evolution of integrated computational and visualisation technologies lead to new opportunities for reinforcing knowledge-based maintenance through collection, processing and provision of actionable information and recommendations for maintenance operators. Providing actionable information regarding both corrective and preventive maintenance activities at the right time may lead to reduce human failure and improve overall efficiency within maintenance processes. Selecting appropriate digital assistance systems (DAS), however, highly depends on hardware and IT infrastructure, software and interfaces as well as information provision methods such as visualization. The selection procedures can be challenging due to the wide range of services and products available on the market. In particular, underlying machine learning algorithms deployed by each product could provide certain level of intelligence and ultimately could transform diagnostic maintenance capabilities into predictive and prescriptive maintenance. This paper proposes a process-based model to facilitate the selection of suitable DAS for supporting maintenance operations in manufacturing industries. This solution is employed for a structured requirement elicitation from various application domains and ultimately mapping the requirements to existing digital assistance solutions. Using the proposed approach, a (combination of) digital assistance system is selected and linked to maintenance activities. For this purpose, we gain benefit from an in-house process modeling tool utilized for identifying and relating sequence of maintenance activities. Finally, we collect feedback through employing the selected digital assistance system to improve the quality of recommendations and to identify the strengths and weaknesses of each system in association to practical use-cases from TU Wien Pilot-Factory Industry 4.0.

Keywords: Maintenance, Digital Assistance Systems, Process Model, Industry 4.0.

1 Introduction

1.1 Digital Assistance in Knowledge-Based Maintenance

Maintenance is a knowledge-intensive process in which the process participants (organizations or (group of) individuals involved in the maintenance process and sub-process(es) either as internal or external stakeholders) create, (re)use, and share specialized professional knowledge, while enriching their implicit and experiential knowledge. Considering maintenance organization as a learnable unit, it encompasses the creation,

acquisition, extraction, storage, retrieval, discovery, application, review, sharing and transfer of the knowledge captured from/within maintenance processes. To this end, Knowledge-Based Maintenance (KBM) continuously supports value generation and facilitates developing and protecting maintenance collective knowledge across maintenance organization, which is enhanced by a variety of data-driven, digital technologies and artificial intelligence (AI) techniques, including advanced statistics, stochastics, real-time computing and analytics, machine learning algorithms, static rule-based or dynamic model-based analytics, and semantic modelling and representations [1],[2]. From a practical point of view, maintenance operators and engineers are frequently associated with a wide range of difficulties due to the increasing complexity of manufacturing systems, in terms of products, processes and systems, namely: i) a wide range of maintenance tasks from diagnosis to repair, ii) increasing complexity of maintenance requirements and iii) a large number of equipment types to maintain [3],[4]. Additionally, they are constantly confronted with situations in which the experiential knowledge of other employees is needed, particularly in the confrontation with new or rarely occurring tasks and circumstances. The challenge that arises with increasing complexity is a shortage of skilled workers and the time required to build up relevant experience [5].

With the digitization of the industry and the recent technological advancements of computing and visualization technologies, the opportunity to access actionable information for maintenance operators and engineers provides additional benefits. The increasing integration of ICT technologies in classical automation as well as a constantly increasing digital database enable them to capture information through a real time interaction [6], [7]. According to our experiential knowledge, almost 90% of maintenance practitioners use a notebook as a tool to obtain information for their maintenance tasks. Nevertheless, hardcopies build the second most common information source. The study participants consider the active support of the diagnosis as well as the availability of information and checklists for the respective process steps to be the most helpful measures during the service visit [8]. Digital assistance systems (DAS) can enhance human performances, depending the degree of digitization, by providing relevant information for a given specific task [9]. Maintenance operators and engineers can capture information through the used device more quickly and more precisely, while they are performing maintenance, inspection or repair tasks [10]. Recent studies show that DAS can increase maintenance practitioners' productivity by 8.5% [3]. However, the reason for selecting a device rather than another is not always trivial and relates to context of application, environmental conditions, the user and the process related requirements [11]. In order to select and make decision on an appropriate device to assist maintenance operators, organisations need to take multiple decision criteria and preferences into account [13]. Research surveys show that companies confront major challenges in implementing digital assistance solutions due to high investment costs and technological issues such as: i) choosing the right hardware, ii) development of a software and realizing a suitable visualisation method and iii) supplying adequate information to improve human performances by providing relevant information regarding both corrective and preventive maintenance [11], [12], [14]. The selection procedures can be challenging due to the wide choice of services (options) available on the market.

Considering the discussion above, this paper presents an approach to improve the maintenance efficiency through DAS using a morphological approach for the proper hardware selection combined with a process-modeling tool providing the adequate information to fulfill the needed maintenance task. The goal of the proposed process model is to systematically identify functionalities of the emerging technologies on the market and apply the functionalities to requirements in order to find appropriate assistance systems for various industrial applications. Therefore, an overview on present digital assistance solutions is given and a morphological approach for the elicitation of derived requirements on digital assistance solutions is presented.

1.2 Digital Assisted Maintenance (DAM)

The emergence of novel wearable technologies (in this paper referred to as a type of DAS) such as smart glasses, smart watches and tablets spurred new concepts of service support systems [9]. DAS combined with Cloud manufacturing concepts provide an opportunity to deal with the increasingly complex maintenance procedures [3], [9]. DAS create the potential to shape new working environments in which modern technology is used to assist workers in activities that are challenging in terms of their cognitive complexity [14]. Via interfaces, corresponding process data are processed and visualized by software components embedded into assistance system to support maintenance operators with relevant information, e.g. by means of head-mounted displays or portable devices. A strong focus of literature is the exploration and identification of application areas for implementing and deploying DAS [5]. To implement DAS, the service-oriented architecture approach has become established. Although innovative technologies, e.g. web services, have already been employed in industrial applications [15], [16], their usage in maintenance support has not been sufficiently well emphasized. A preliminary chronological market and literature analysis with regard to suitability and industrial applicability (i.e. technology readiness) of DAS, in particular wearable devices, is shown in Fig.1.

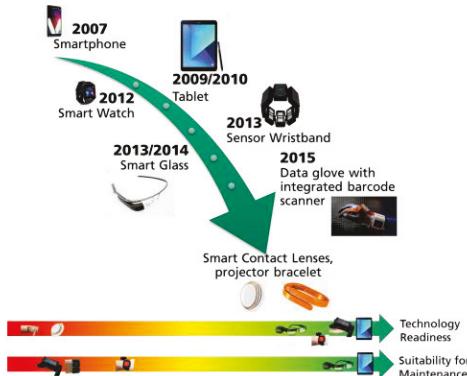


Fig. 1. Overview of digital assistance systems on the market and their market entrance.

As a result, the four most common DAS in industrial application are: industrial tablets, smart watches, smart phones and head mounted displays [12], [17], [18], [19]. While the pros and cons of handheld devices (industrial tablets, smart watches, smart

phones) are well known and elaborated in literature, the potential of head-mounted displays are disputed. The most value-creating functionalities of head mounted displays lie in information provision, environmental identification and tracking [6]. The opportunity to access information hands free provides additional benefits. However, due to various technical limitations and challenges, such as wear comfort or poor wireless network connections, the question of usefulness in maintenance still arises.

2 Selection Methodology

This section explains the methodology of the developed model to select proper DAS for maintenance tasks. The proposed model builds on three integrated elements (see Fig. 2): i) Morphological Approach, ii) Application Layer and iii) Device Selection Layer.

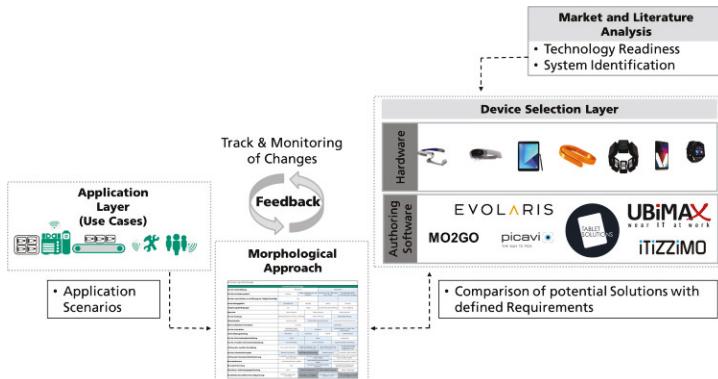


Fig. 2 3-layer model for selecting proper digital assistance systems

The first element represents a morphological box, which has been developed to facilitate and optimize the selection of suitable DAS. The second element represents the application domain. The application layer provides the individual user-specific system requirements as well as application scenarios (i.e. describing and representing maintenance activities). Subsequently, the system and hardware requirements resulting from i) the predefined parameters of the requirement morphology and ii) from the application level are evaluated and, according to their overall systemic meaning, compared with the potential technology solutions. Algorithms and correlation-analyses within this system are used to ultimately map the requirements to existing digital assistance solutions. Using the proposed approach, a (combination of) DAS is selected and linked to maintenance activities. Finally, we collect feedback through employing the selected digital assistance system to improve the quality of recommendations and to identify the strengths and weaknesses of each system in association to practical use-cases.

2.1 Morphological approach

In order to facilitate and optimize the selection of suitable DAS for supporting maintenance operations in manufacturing industry, a morphological approach has been developed. A Morphological Analysis (MA) represents a method for systematically structuring and analyzing a set of relationships contained in multi-dimensional, non-quantifiable problem complexes [20], [21]. MA usually consists of three steps. First,

the problem complexity is categorized into several dimensions. Second, all possible conditions (also referred to as parameters) to each dimension are identified. These parameters represent the characteristics of each dimension. Finally, a morphological matrix is developed based on the identified dimensions and their assigned condition parameters [22]. Figure 3 depicts a morphological matrix, which contains a collection of identified features that are critical to selecting an assistance system. Key features for an adequate assistance system can be categorized into three groups: i) requirements regarding the application (software): How and to what extent maintenance information is presented to maintenance operators and engineers towards increasing their performance in an affordable manner? ii) requirements regarding the information system: How and to what extent maintenance information is tailored to the application? iii) requirements regarding the hardware: which hardware should be applied for the selected case?

Requirement Morphology				
Dimensions	Parameters			
Type of support	Physically		Information	
Type of assistance systems	Stationary	Mobile installation (for example on a notebook)	Hand device (for example tablet or smart phone)	Wearable (smart glasses, smartwatch or other)
Two hands are needed to carry out the activity	No		Yes	
Application	Maintenance	Assembly	Logistics	Others
Environmental conditions	wet	dusty	fluctuating lighting conditions	noisy environment
Operator	Gloves required	Glasses required		Helmet required
Type of use	One-time use (training, education)		Selective use	Regular use
Data transfer	Local via cable	Wireless via private network	Wireless via public network	
Human Machine Interaction	Unimodal		Multimodal	
Type of interaction	Monological (without possibility of interaction)		Dialogical	Communicating (for example via chat functionality)
Support provided	Information	Instruction	Interference	Engagement Documentation
Method of providing information	Visually	Auditive	Kinesthetic	
Type of Visual Information Presentation	On-Screen Display	On-Site Projection		Augmented Reality Visualization
Scope of visual presentation	No visual information	Simple presentation (texts, images or markings)	Multi media presentation on (videos, animations, etc.)	Advanced presentation (selected drawing formats, etc.)
Type of information input	Manual (via actuator)	Verbal (via voice control)	Gesture recognition	Automatic (via sensors)
Scope of user customization	Predefined configuration with regard to input and output		Individual configuration regarding output possible	Individual configuration with regard to input and output possible
User categories	No user groups possible		Different user groups possible	
User recognition	None		User login and initialization	Automatic detection and initialization of the user
Situation and motion detection	None	Using motion sensors	Using optical sensors	Other technology
Flexibility (with regard to reconfiguration)	Substantial adjustments necessary	Adjustments necessary (to be carried out by specialists)	Adjustments necessary (by specialist on site)	Small adaptation necessary (by the user on site)

Fig. 3 Morphological Matrix with defined dimensions and parameters

Based on a literature review and collection of manufacturers' data, we derived and determined 20 relevant criteria to assess the digital assistance system requirements. These requirements represent the dimensions within the morphological matrix. To further systematize the requirement analysis, the requirement elements are classified below using a morphological matrix in Fig. 3. This morphological matrix contains a collection of general condition parameters of the presented requirements. Based on the individual user-specific system requirements and the application scenario, the characteristic parameters can be identified. Thereby, the requirements are highlighted in color and optional requirements are shaded in color. It should be noticed that each answer can affect more than one choice regarding the hardware, software and visualization method. For the selection of a suitable digital assistance system a decision hierarchy

needs to be constructed [22]. The underlying algorithm is based on Analytic Hierarchy Process (AHP) and fuzzy TOPSIS method principles [22]. An AHP method was applied to calculate the criteria priority weights, while fuzzy TOPSIS is used to evaluate and select a proper (combination of) digital assistance system(s).

2.2 Device Selection Layer

The Device selection layer represents the technology database and includes hardware and authoring software solutions. Due to the novelty of the topic, we had access to only a few practitioners with real-life implementations of DAS, especially smart glasses in this matter. For this reason, we have decided to elicit the functionalities from systematic literature review and market analysis. Based on ISO16290 the Technology Readiness Level of the emerging technologies on the market has been rated (see Fig. 1).

3 Case Study: An Explanatory Process Model for DAM

To reveal the functionality of the developed process model a maintenance scenario has been developed within the TU Wien Pilot-Factory Industry 4.0. Up to now, maintenance processes on the Universal Robot (UR5 laboratory robot) have only been carried out by experts. A proper DAS should guide workers step by step through the maintenance activities on the machine. Based on the developed selection model (cf. Section 2), the use of smart glasses is recommended. The chosen underlying software tool is called MO²GO, a Process Modeling Tool developed by the Fraunhofer Institute for Production Systems and Design Technology IPK.

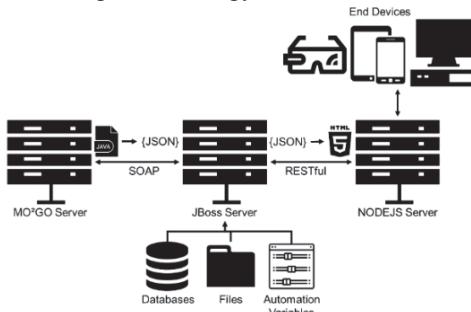


Fig. 5. Schematic software architecture for a context sensitive digital assistance system.

To model business processes, the method of integrated enterprise modeling (IEM) was developed in the 1990s at the Fraunhofer IPK [23]. The application of the IEM supports the description of business processes and their interactions with description elements of companies, such as organization, system, product or control. It is compatible with DIN EN ISO 19440 "Enterprise Integration - Constructs for Enterprise Modelling" and describes four element classes that can be related by five connection types. Table 1 shows a selection of element classes and connection types which are needed to model maintenance processes. The graphical modeling tool MO²GO[24], also developed at Fraunhofer IPK, is well suited to model the maintenance processes and forms the basis for the implementation of DAS[25]. MO²GO supports the XML (eXtensible

Markup Language) exchange format, which is suitable for exchanging data between different applications. For the process step representation in a graphical user interface (GUI) of a digital assistance system, MO²GO offers an interface to provide the XML format of the process model as a JAVA object representation. The elements and their connections are then converted to JSON format and interpreted by an application interface (API) to link resources, generate context sensitive instructions and to initialize support functions on the maintained system during the various process steps. This JSON representation is then transformed to the web-capable HTML5 format in which JAVA Script is embedded to realize human-machine-interaction.

Table 1. Excerpt of IEM classes and connection types used for maintenance process modeling.

Objects and linkage types	Description
Product	All objects that are changed by activities during a field service deployment, e. g. the product “machine tool with failure” (start condition) is changed by the activity “performing service deployment” towards the product “machine tool without failure” (final condition)
Activity	Changes the condition of a product
Resource	All needed objects necessary to perform an activity, e. g. web service call to invoke a test routine on the machine tool
Sequence	Activities are performed successively
Parallel branching	Activities are performed parallel; parallel activities have to be completed before the next activity can be started
Loop	The activity in the loop will be performed until the condition for starting the next activity is met

Figure 6 shows a scenario for the exchange of gripper jaws.

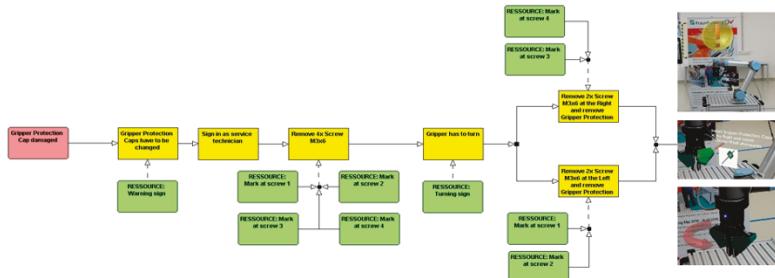


Fig. 6 Pictorial representation of a need for action and textual explanation of the activity combined with a pictorial representation of the tool and the object to be exchanged.

The maintenance operator is assisted by step-by-step instructions through virtual information directly on the work object. The user interface has been kept simple i.e. users

see a complete virtual model of the equipment and the needed information to fulfil the maintenance task to the right. The MO²GO model is used to provide logic and information for the augmented reality (AR) based assistance system and to guide the worker through eight process steps.

4 Conclusion and Outlook

The presented approach can serve as guidance for the strategic evaluation of digital assistance solutions supporting maintenance processes. Combined with the proposed process-modeling tool the assistance system can provide the needed information to improve the maintenance efficiency. Since the proposed approach is currently a prototype, it encompasses some limitations that necessitate further research. First, the underlying decision hierarchy is based on experimental knowledge of experts and has been only validated through the proposed use case. In order to improve the quality of recommendations we need to collect feedback through employing the proposed approach to further practical use-cases. Second, the key information regarding hard- and software of the DAS on the market is extracted manually. By using various web crawling and web analytic techniques, including automated text- and web-mining methods, information can be extracted from documents such as product manuals and patent documents dynamically to identify the key features of existing products and technologies.

Acknowledgement

The authors would like to acknowledge the financial support of the European Commission provided through the H2020 project EPIC under the grant No. 739592. The TU Wien Pilot Factory Industry 4.0 has been partly funded by the public through the Austrian Research Promotion Agency (FFG) and several private industrial firms – our partners in the project.

References

- Ansari, F.: Meta-Analysis of Knowledge Assets for Continuous Improvement of Maintenance Cost Controlling. Faculty of Science and Technology, University of Siegen (2014).
- Nemeth, T., Ansari, F., Sihn, W., Haslhofer, B., Schindler, A.: PriMa-X: A Reference Model for Realizing Prescriptive Maintenance and Assessing its Maturity Enhanced by Machine Learning. Procedia CIRP, Vol. 72, pp. 1039-1044. (2018).
- Glawar, R., Karner, M., Nemeth, T., Matyas, K., Sihn, W.: An Approach for the Integration of Anticipative Maintenance Strategies within a Production Planning and Control Model. Procedia CIRP 67 46 – 51, (2018).
- Hao, Y., & Helo, P.: The role of wearable devices in meeting the needs of cloud manufacturing: A case study. Robotics and Computer-Integrated Manufacturing, 45. Jg., S. 168-179. (2017).
- Kernchen, A., Jachmann, D., Adler, S.: Assistenzsysteme für die Instandhaltung und Störungsbehebung. 21. Magdeburger Logistik Tage. Logistik neu denken und gestalten. S.195. (2016).
- Niemöller, C., Metzger, D., Fellmann, M., Özcan, D., Thomas, O.: Shaping the future of mobile service support systems-ex-ante evaluation of smart glasses in technical customer service processes. Informatik 2016, (2016).
- Erkoyuncu, J. A., del Amo, I. F., Dalle Mura, M., Roy, R., Dini, G.: Improving efficiency of industrial maintenance with context aware adaptive authoring in augmented reality. CIRP Annals 66.1. 465-468. (2017).

8. Uhlmann E., Ruae N., Geisert C.: Unterstützungspotenziale der Automatisierungstechnik im technischen Kundendienst. Summary of an explorative survey on best practices in field service. Berlin: Fraunhofer IPK, (2013).
9. Mourtzis, D., Zogopoulos, V., Vlachou, E.: Augmented reality application to support remote maintenance as a service in the Robotics industry. Procedia CIRP 63: 46-51. (2017).
10. Neges, M., Wolf, M., Abramovici, M.: Secure access augmented reality solution for mobile maintenance support utilizing condition-oriented work instructions. Procedia CIRP, 38, 58-62. (2015).
11. Palmarini, R., Erkoyuncu, J., Rajkumar, R.: An innovative process to select Augmented Reality (AR) technology for maintenance. Procedia CIRP 59: 23-28 (2017).
12. Palmarini, R., Erkoyuncu, J. A., Roy, R., Torabmostaedi, H.: A systematic review of augmented reality applications in maintenance. Robotics and Computer-Integrated Manufacturing 49: 215-228. (2018).
13. Hold, P., Erol, S., Reisinger, G., & Sihn, W.: Planning and Evaluation of Digital Assistance Systems. Procedia Manufacturing 9:143-150. (2017).
14. Reisinger, G., Komenda, T., Hold, P., & Sihn, W.: A Concept towards Automated Data-Driven Reconfiguration of Digital Assistance Systems. Education & Training 2351: 9789. (2018).
15. Hohwieler E., Geisert C.: Intelligent Machines Offer Condition Monitoring and Maintenance Prediction Services. In: Teti R, editor. Proceedings of the 4th CIRP International Seminar on Intelligent Computation in Manufacturing Engineering (CIRP ICME '04). 30 June - 2 July 2004, Sorrento, Italy; pp. 599-604. (2004).
16. Hohwieler E., Berger R., Geisert C.: Condition Monitoring Services for e-Maintenance. In: Zaremba M, Saslaidek J, Erbe HH, editors. A proceedings volume from the 7th IFAC Symposium, Gatineau, Québec, Canada, 6-9 June 2004. Oxford: Elsevier pp. 239-244. (2005).
17. Ziegler, J., Heinze, S., Urbas, L.: The potential of smartwatches to support mobile industrial maintenance tasks. Emerging Technologies & Factory Automation (ETFA), IEEE 20th Conference on. IEEE, (2015).
18. Bokrantz, J., Skoogh, A., Berlin, C., & Stahre, J.: Maintenance in digitalised manufacturing: Delphi-based scenarios for 2030. International Journal of Production Economics, 191, 154-169. (2017).
19. Hold, P., Ranz, F., Hummel, V., Sihn, W.: Durchblick im Variantendschungel: visuelle Assistenzsysteme als Flexibilitätshebel auf dem Shop Floor (2015).
20. Ritchey, T.: Modeling alternative futures with general morphological analysis. World Future Review, 3(1), 83-94. (2011).
21. Ritchey, T.: Problem structuring using computer-aided morphological analysis. Journal of the Operational Research Society, 57(7), 792-801. (2006).
22. Im, K., Cho, H.: A systematic approach for developing a new business model using morphological analysis and integrated fuzzy approach. Expert Systems with Applications, 40(11), 4463-4477. (2013).
23. Spur, G.; Mertins, K.; Jochem, R.: Integrated Enterprise Modelling. Berlin, Wien, Zürich: Beuth. (1996).
24. Mertins K, Jaekel FW.: MO²GO: User Oriented Enterprise Models for Organisational and IT Solutions. In: Schmidt G, Mertins K, Bernus P, editors. Handbook on architectures of information systems. Berlin, New York: Springer p. 649-663. (2006).
25. Uhlmann, E.; Geisert, C.; Ruae, N.; Gabriel, C.: Situation Adapted Field Service Support Using Business Process Models and ICT Based Human-Machine-Interaction. Procedia CIRP 47, p. 240-245. (2016).

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Detection of Directed Connectivities in Dynamic Systems for Different Excitation Signals using Spectral Granger Causality

Christian Kühnert¹, Christian Frey¹ and Ruben Seyboldt¹

Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB
Fraunhoferstraße 1, 76131 Karlsruhe, Germany
{christian.kuehnert, christian.frey, ruben.seyboldt}@iosb.fraunhofer.de

Abstract. Industrial plants usually consist of different process units which are strongly cross-linked to each other. This leads to the point that a voluntary or involuntary change in one unit (e.g. changing some process control parameter or having a malfunctioning value) can lead to unexpected results in another process unit. Hence, knowing which are the causing and which are the effecting process variables is of great interest. Still, depending on the underlying process and the characteristics of the excitation signal, directed connectivities can or can not be detected. Therefore, in this paper several types of dynamic SISO systems and excitation signals are defined for which a directed connectivity from input to output signal should be detected and from output to input should not be detected. As a method for the detection of directed influences Spectral Granger Causality is used, which has been extended with a surrogate-based significance test. This test is used to define if a directed influence exists from one process variable to another.

Keywords: Spectral Granger Causality · Detection of Directed Connectivities · Time Series Analysis.

1 Introduction

Process control systems at production plants usually consist of a large number of process variables, while the interconnectivity of the variables is not always directly evident. Hence, due to the interconnectivity, if some change, voluntary or on purpose is performed on one unit, this can lead to unwanted effects at another unit. Therefore, it is of great interest to understand which variable has a significant influence on another variable.

For the automatic detection of directed connectivities in time series exists already a wide variety of methods, which are mainly developed for the use in neuroscience (e.g. [3] or [1] for reviews) or for the analysis of econometric data [9]. One of the first methods developed, was done by Granger [8], being called the Granger Causality. This method uses two vector autoregressive functions and, by comparing their residual sum of squares, the method tells if one variable causes the other or not. The original approach, taking place in the time domain,

was extended by Geweke [7] into the spectral domain, having the advantage to select specific frequencies for analysis. In 2000 Schreiber [14] developed a method called Transfer Entropy, which measures the amount of information transferred from one random process to another. In recent research, Transfer Entropy has been extended by contains several extensions like Partial Transfer Entropy [11] or Symbolic Entropy [13]. Bauer [2] proposes a Nearest-Neighbor approach for cause-effect analysis. In [12] different methods for the detection of significant directed influences were developed and compared on several benchmarks, consisting of simulated dynamic systems data, biosignals and on disturbances from a glass forming process. Kaminski [10] proposes the estimation of directed transfer functions.

This aim of this paper to investigate under which circumstances it is possible to detect directed influences in measurements, depending on the excitation signal as well as the underlying dynamic systems. As specific detection method Spectral Granger Causality [7] is used, which is extended with a surrogate-based significance test. In difference to [12], which already defines benchmark processes for the detection of causal dependencies, the current paper focuses more on the excitation signal characteristics.

The paper is structured as follows: Section 2 introduces how directed connectivities can be detected in time series and how Spectral Granger Causality is applied. Additionally, the surrogate-based calculation of the significance threshold is explained. Section 3 describes the defined input signals and dynamic systems for benchmarking, while section 4 discusses the results. Finally, section 5 gives a summary and some ideas for future research.

2 Detecting directed connectivities in time series

2.1 Bivariate Spectral Granger Causality

The concept of Granger causality (GC) has been originally introduced in the field of economics by Clive Granger in 1969 [8] who used it to determine the relationships of different econometric models. The basic concept of bivariate GC can be explained by assuming the two time series $u[k] \in \mathbb{R}$ and $y[k] \in \mathbb{R}$ with $k = 1, \dots, K$ samples. In that case, the causal connectivity $u \rightarrow y$ is assumed to exist if past values from $u[k]$ and $y[k]$ result in a higher forecast accuracy for $y[k]$ than using only past values from $y[k]$. Mathematically, this is evaluated by comparing two linear vector autoregressive models, while the first one only contains past values of $y[k]$, called the restricted model, and the other one containing past values of $u[k]$ and $y[k]$, called the unrestricted model.

Furthermore, Granger Causality can be easily extended into the multivariate case, while good explanations can be found e.g. in [16] or [4]. Since the developed benchmarks in section 3 compare always one input against one output, for simplicity, multivariate GC will not be explained in this paper.

GC in the time domain: Checking if u causes y or y causes u , is in the time domain is done by comparing two linear vectorautoregressive (VAR) models.

The two VAR models are defined as

$$u[k] = \sum_{j=1}^n a_{uu}[j] \cdot u[k-j] + \sum_{j=1}^n a_{uy}[j] \cdot y[k-j] + e_u[k], \quad (1)$$

$$y[k] = \sum_{j=1}^n a_{yy}[j] \cdot y[k-j] + \sum_{j=1}^n a_{yu}[j] \cdot u[k-j] + e_y[k] \quad (2)$$

containing the residual covariance matrix being defined as

$$\Sigma = \begin{pmatrix} \Sigma_{uu} & \Sigma_{uy} \\ \Sigma_{yu} & \Sigma_{yy} \end{pmatrix} \quad (3)$$

In 1, 2 n is the model order, $a_{uu}, a_{uy}, a_{yu}, a_{yy} \in \mathbb{R}^n$ contain the model coefficients and $e_u[k], e_y[k] \in \mathbb{R}$ define the residuals. Finally GC checks the coefficients in a_{yu} (respectively a_{uy}). If these are significantly different from zero, it is assumed that u causes y (respectively y causes u). Usually, this is done by comparing the squared-sum of residuals of e_u (respectively e_y) with and without taking into account the influencing variable y (respectively u).

GC in the frequency domain The advantage when working in the frequency domain compared to the time domain is that causal connectivities can be tied to specific frequency bands and one gets better insights in the data. The methodology has been explained in detail in [7] and the main steps are given here for completeness. The Fourier Transformation of the equations 1 and 2 can be written in the following set of equations:

$$\begin{pmatrix} A_{uu}(f) & A_{uy}(f) \\ A_{yu}(f) & A_{yy}(f) \end{pmatrix} \begin{pmatrix} u(f) \\ y(f) \end{pmatrix} = \begin{pmatrix} e_u(f) \\ e_y(f) \end{pmatrix} \quad (4)$$

with $u(f)$ and $y(f)$ are the Fourier transformed time series from $u[k]$ and $y[k]$ and $e_u(f), e_y(f)$ are the Fourier transformations of $e_u[k]$ and $e_y[k]$. The components of A are then transformed as

$$A_{uu}(f) = 1 - \sum_{i=1}^n a_{uu}(n) e^{(-i2\pi f n)} \quad (5)$$

$$A_{uy}(f) = - \sum_{i=1}^n a_{uy}(n) e^{(-i2\pi f n)} \quad (6)$$

which counts analogous for $A_{yu}(f)$ and $A_{yy}(f)$. Finally, equation 4 can be rewritten as

$$\begin{pmatrix} u(f) \\ y(f) \end{pmatrix} = \begin{pmatrix} H_{uu}(f) & H_{uy}(f) \\ H_{yu}(f) & H_{yy}(f) \end{pmatrix} \begin{pmatrix} e_u(f) \\ e_y(f) \end{pmatrix} \quad (7)$$

with $H(f)$ defining the transfer function matrix. Following Geweke [7], under the assumption that the covariance $\Sigma_{uy} = 0$, the auto spectrum $S_{uu}(f)$ for the time series $u[k]$ can be derived as

$$S_{uu}(f) = H_{uu}(f)\Sigma_{uu}H_{uu}(f)^* + H_{uy}\Sigma_{yy}H_{uy}(f)^*. \quad (8)$$

The asterisk in equation 8 defines the transposed and complex conjugated transfer function. According to Seth [15], equation 8 can finally be divided into an intrinsic part, namely $H_{uu}(f)\Sigma_{uu}H_{uu}(f)^*$ and a causal part, namely $H_{uy}\Sigma_{yy}H_{uy}(f)^*$. Hence, the Granger Causality for each frequency can be calculated as

$$f_{u \rightarrow y}(f) = \ln \left(\frac{|S_{uu}(f)|}{|S_{uu}(f) - H_{uy}\Sigma_{yy}H_{uy}(f)^*|} \right).$$

Finally, the causal strength $\mathcal{F}_{u \rightarrow y}$ is calculated by integrating over the complete frequency spectrum being defined as

$$\mathcal{F}_{u \rightarrow y} = \frac{1}{2\pi} \int_0^{2\pi} f_{u \rightarrow y}(f) df \quad (9)$$

2.2 Threshold

The in equation 9 defined causal strength $\mathcal{F}_{u \rightarrow y}$ is not bounded, meaning that from the bare value it is not possible to tell if a causal dependency is really significant or not. Therefore, a threshold needs to be calculated each time an input u is tested against a possible output y . Following Choudhury [5] a surrogate time series needs to be calculated for u , while surrogate means that the phase coupling is removed but the signal keeps the same power spectrum. In other words, all causal information is removed from the signal. To calculate the surrogate of u the following steps need to be performed

$$u_{\text{FFT}} = \text{FFT}(u)$$

$$u_{\text{FFT}}^{\text{surr}} = \begin{cases} u_{\text{FFT}}[k] & k = 1, N/2 + 1 \\ u_{\text{FFT}}[k]e^{j\Phi_{k-1}} & k = 2, \dots, N/2 \\ u_{\text{FFT}}[k]e^{j\Phi_{k-1}} & k = (N/2 + 2), \dots, N \end{cases}$$

$$u^{\text{surr}} = \text{IFFT}(u_{\text{FFT}}^{\text{surr}})$$

with FFT being the Fourier and IFFT being the Inverse Fourier Transform. In that case N describes the number of samples and $\Phi_n \in [0, \dots, 2\pi]$ with $k = 1, \dots, (N/2 - 1)$ is a random phase value. The final threshold is derived in terms of a 3σ test being defined as

$$\mathcal{F}_{u \rightarrow y}^{\text{Threshold}} = \mu^{\text{surr}} + 3\sigma^{\text{surr}}$$

with

$$\mu^{\text{surr}} = \frac{1}{M} \sum_{k=1}^M \mathcal{F}_{u^{\text{surr}} \rightarrow y}, \quad \sigma^{\text{surr}} = \sqrt{\frac{1}{M} \sum_{m=1}^M (\mathcal{F}_{u^{\text{surr}} \rightarrow y} - \mu^{\text{surr}})^2}.$$

and M being the number of surrogate trials. If the outcome indicates $\mathcal{F}_{u \rightarrow y} > \mathcal{F}_{u \rightarrow y}^{\text{Threshold}}$, the found causal dependency is defined as being significant.

3 Benchmarks

For the detection of directed connectivities in time series two things are important, namely the characteristics of the excitation signal and the underlying process behavior. Hence, this section proposes several possible input signals (section 3.1) and several dynamic SISO systems (section 3.2). Next, the Spectral Granger Causality is used to detect the input and output signal for each pair.

3.1 Analyzed Excitation Signals

As excitation signals white noise, a sinusoid, the sawtooth wave, an impulse train and a time series based on a random walk are used. All signals are shown in figure 1 in the time domain as well as its power spectrum. For analysis, each signal consists in summary of $N = 1000$ samples. The details of the excitation signals are as follows:

White Noise - A time series that consists of white noise means to have a sequence of uncorrelated random variables with constant mean μ and variance σ^2 . In the following, the input time series $u_{\text{wn}}[k] \in \mathbb{R}$ is modeled as a stochastic process with $\mu = 0$ and $\sigma^2 = 1$.

Sinusoid - A sinusoid can be seen as a prototype of a periodic disturbance, resulting e.g. from poorly tuned PI-controllers. For the input series a sinusoid of the form $u_{\text{sin}}[k] = \sin(\omega \cdot k) \in \mathbb{R}$ is used with an angular frequency of $\omega = 2\pi \cdot 0.1$.

Sawtooth Wave - This time series can be interpreted as some sort of a drift e.g. when sensors are slowly polluting. For the sawtooth wave the input series $u_{\text{sw}}[k] \in \mathbb{R}$ is defined as $u_{\text{sw}}[k] = \text{frac}(\frac{k}{T} + \Phi)$ with a period of $T = 100$ and the phase $\Phi = 0$ and frac being the fractional part defined as $\text{frac} \equiv x - [x]$.

Impulse train - Having so-called impulse or spike train means that e.g. an inert gas or fluid injection into a process at a predefined cycle occurs. Therefore, the input time series $u_{\text{it}}[k] \in \mathbb{R}$ is defined as $u_{\text{it}}[k] = \sum_{k=0}^{N-1} \delta[n - kK]$ with $N|K$, δ being a Dirac impulse, $N \in \mathbb{R}$ representing the length of the time series and $K \in \mathbb{R}$ the period. In the following the period K is set to 100.

Random Walk - The time series of a random walk is defined as a process where the value at sample point $[k]$ is composed of the past value $[k - 1]$ plus an error term defined as white noise. In this paper the random walk is used to investigate how used methods behave on low-frequent changes in a process e.g. when having a fluctuation of some concentration in a fluid. Therefore, the input time series $u_{\text{RW}}[k] \in \mathbb{R}$ is defined as $u_{\text{RW}}[k] = u_{\text{RW}}[k - 1] + \epsilon[k]$ where $\epsilon[k]$ is a white-noise sequence with $\mu = 0$ and $\sigma^2 = 0.1$.

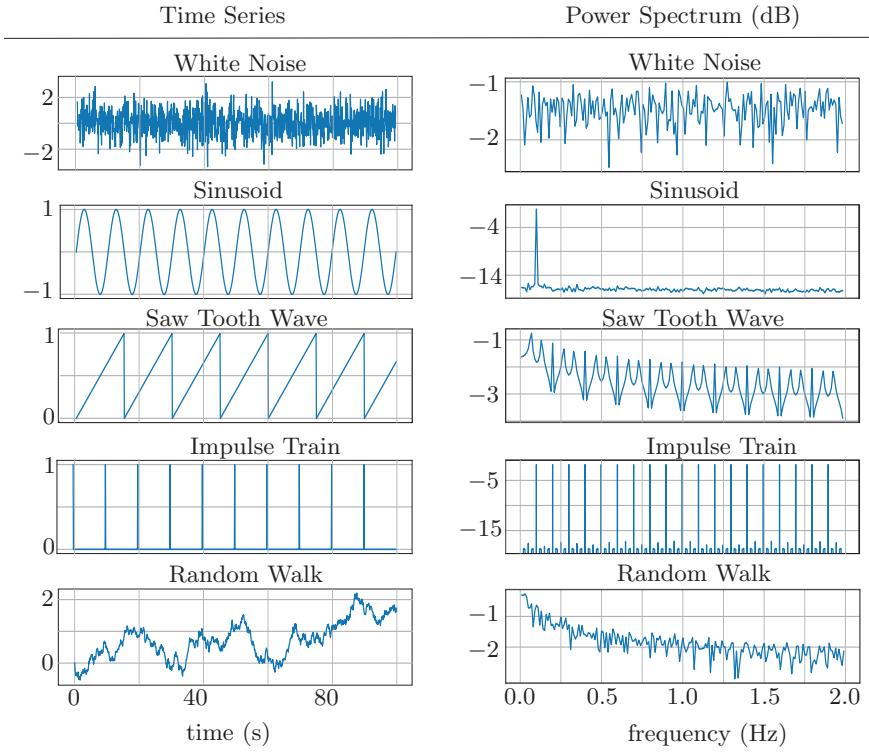


Fig. 1. Investigated excitation signals in the time domain and their corresponding power spectra.

3.2 Dynamic systems

Figure 2 shows the selected dynamic systems which are tested in combination with the prior shown excitation signals. In detail, the systems consist of a dead time, a low-pass filter, a nonlinearity and finally a resonant second order system. In detail, the systems are described as follows:

Dead Time - In this benchmark, the excitation signal is shifted by 10 samples. No dynamic system is added between input and output signal. Hence, this responds to the most simple case for the detection of directed connectivities from one signal to another.

Low-pass filter - The low pass filter with the time constant $T = 1\text{ s}$ represents the most basic system for the detection of input and output signal. In process technology low-pass filter are e.g. fluid tanks or pipes which tend to attenuate a disturbance and hence making it sometimes complicated to track back the disturbance propagation path. This benchmark is mainly used to investigate the behavior regarding the defined input signals in section 3.1.

Nonlinear system - In this process a sinusoid is taken from the intermediate output signal $y_1(t)$. Depending on the amplitude of the excitation signal, the sinusoid will have a strong impact on the resulting output signal. The main purpose is to determine for which input signals the methods can still determine the input and output signals and their parameters.

Resonant system - This benchmark represents a classic mass-spring-damper system. Like for the other systems the time constant is set to $T = 1\text{ s}$, while the dimensionless damping ratio is set to $\xi = 0.05$.

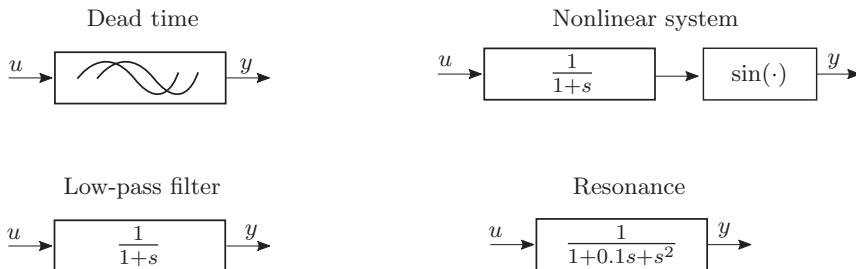


Fig. 2. Used transfer functions for the validation of the detection of directed influences.

4 Results

For analysis, each dynamic systems wax excited with the different input signals and the spectral Granger causality was used for the detection of directed influences from $u \rightarrow y$, with results shown in figure 4, and from $y \rightarrow u$, where the results are shown in figure 4. If a directed influence has been found, the corresponding box contains a checkmark, otherwise it contains a cross. In the following a summary is given by following corresponding to the defined benchmarking dynamic systems.

Dead time: In that use case, consisting of a simple time shift, for all input signals, the directed dependencies from $u \rightarrow y$ are detected and defined as being significant. Nevertheless, for the input signal u_{sin} and u_{imp} a false positive directed influence has been found pointing from $y \rightarrow u$. The explanation is straight forward, since the impulse train as well as the sinusoid are cyclic excitations signals, hence having only a time shift in the signals, it is not possible to distinguish input from output signal .

Dead time						Nonlinear System					
y_{out}	✓	✓	✓	✓	✓	y_{out}	✗	✓	✗	✓	✓
	u_{wn}	u_{sin}	u_{st}	u_{imp}	u_{rw}		u_{wn}	u_{sin}	u_{st}	u_{imp}	u_{rw}
Low-pass filter						Resonance					
y_{out}	✓	✓	✗	✓	✗	y_{out}	✓	✓	✗	✗	✓
	u_{wn}	u_{sin}	u_{st}	u_{imp}	u_{rw}		u_{wn}	u_{sin}	u_{st}	u_{imp}	u_{rw}

Fig. 3. Results of the benchmarks when testing for directed influences $\mathcal{F}_{u \rightarrow y}$

Dead time		Low-pass filter		Resonance		Nonlinear System	
u_{wn}	✗	u_{wn}	✗	u_{wn}	✗	u_{wn}	✓
u_{sin}	✓	u_{sin}	✓	u_{sin}	✗	u_{sin}	✗
u_{st}	✗	u_{st}	✗	u_{st}	✗	u_{st}	✓
u_{imp}	✓	u_{imp}	✗	u_{imp}	✗	u_{imp}	✗
u_{rw}	✗	u_{rw}	✗	u_{rw}	✓	u_{rw}	✓
y_{out}		y_{out}		y_{out}		y_{out}	

Fig. 4. Results of the benchmarks when testing for directed influences $\mathcal{F}_{y \rightarrow u}$

Low-pass filter: Regarding the low pass filter, u_{wn}, u_{sin} and u_{imp} detect the correct directed connectivity. The saw tooth and random walk, both having a similar power spectrum (see figure 1) are not detected. The reason is that the low-pass filter has to too much attenuation, resulting in an output signal which has already too much information about itself in past values. Hence, in terms of Granger Causality, this results in a non-significant information gain for u_{rw} . The only excitation signal leading to a connectivity from $y \rightarrow u$ is the sinusoid. Like for the dead time benchmark, the reason is that the sinusoid is cyclic.

Nonlinear system: Adding an additional sinusoid as a non-linearity to the low-pass filter in the prior benchmark changes the results of the detected directed influences significantly. u_{wn} does no longer detect the connectivity $u \rightarrow y$, while the u_{rw} is detecting it. The two excitation signals u_{sin} and u_{imp} behave like without non-linearity. Regarding the directed, causal wrong influence $y \rightarrow u$ the excitation signals u_{wn}, u_{st} and u_{rw} detect this influence. Only the u_{sin} and u_{imp} correctly define the influence as not significant.

Resonance: Detecting $u \rightarrow y$ in the spring-mass-damper benchmark is only possible with u_{wn} , u_{sin} and u_{rw} . When having as excitation the signals u_{imp} and u_{st} , spectral Granger Causality assumes that there is neither a directed influence from $u \rightarrow y$ nor from $y \rightarrow u$. Furthermore, except for u_{rw} none of the excitation signals detect in a wrong causal influence $y \rightarrow u$.

5 Summary

The results showed when using spectral Granger Causality, the detection of directed influences in time series depends the exciting signal as well as on the underlying dynamic system. Regarding the excitation signals, for none of the signals it was possible to detect for all four dynamic systems the correct directed influence $u \rightarrow y$, while at the same time never detecting a wrong influence $y \rightarrow u$. Hence, when using Granger Causality, detected or not detected directed influences in data always need to be questioned in terms of the excitation as well as in terms of the underlying process behavior. Still, this method can be of great help to generate first a understanding of the influences variables have onto each other in a data set, since no always, but most of the times Granger Causality detected the correct dependency.

In terms of the development of benchmarks, there is a variety of future research. Questions that arise are the impact of noise in the data or how a directed influence can still be detected if variables having a common cause. Regarding Granger Causality, it can be evaluated, in which cases it is possible to differentiate between direct and indirect influences, e.g. when using the multivariate Granger Causality. Additionally, the benchmarks should be used to compare several methods like Transfer Entropy with its extensions or the estimation of Directed Transfer Functions.

6 Acknowledgements

This work was developed in the Fraunhofer Cluster of Excellence "Cognitive Internet Technologies".

References

1. Bastos, A. and Schoffelen, J.: A Tutorial Review of Functional Connectivity Analysis Methods and Their Interpretational Pitfalls, *Frontiers in Systems Neuroscience*(9), pp. 175, (2016)
2. Bauer, M.: Data-driven Methods for Process Analysis. University College London, PhD Thesis, (2005)
3. Blinowska, K.: Review of the methods of determination of directed connectivity from multichannel data, *Medical & Biological Engineering & Computing* (49), pp. 521 - 529, (2011)
4. Blinowska, K. et al.: Granger causality and information flow in multivariate processes, *Phy. Rev. E.*(70), no.p. 4, (2004)

5. Choudhury, A.A.S. and Shah, S.L. and Thornhill, N.F.: Diagnosis of Process Non-linearities and Valve Stiction: Data Driven Approaches, Advances in Industrial Control, Springer, (2008)
6. Spectral connectivity toolbox: "https://github.com/Eden-Kramer-Lab/spectral_connectivity/", last access August 2018
7. Geweke, J., Measurement of linear dependence and feedback between multiple time series. Journal of American Statistical Association, vol. 77. pp 304–313 (1982)
8. Granger, C. W.J.: Investigating Causal relations by Econometric Models and Cross-Spectral Methods, Econometrica 37, (1969)
9. Heckman, J.: Econometric causality, International statistical review(76), pp. 1–27, (2008)
10. Kaminski M., and Blinowska K.J.: Directed Transfer Function is not influenced by volume conduction—inexpedient pre-processing should be avoided, Frontiers in Computational Neuroscience. (8), (2014)
11. Kugiumtzis, D.: Partial transfer entropy on rank vectors, The European Physical Journal Special Topics (222), pp. 401–420, (2013)
12. Kühnert, C., et al.: Methoden zur datengetriebenen Formulierung und Visualisierung von Kausalitätshypothesen. at - Automatisierungstechnik Methoden und Anwendungen der Steuerungs-, Regelungs- und Informationstechnik, 60(10), pp. 630–640. , (2012)
13. Staniek, M. and Lehnertz, K.: Symbolic transfer entropy: inferring directionality in biosignals, Biomedizinische Technik/Biomedical Engineering (54), pp.323–328 , (2009)
14. Schreiber, T.: Measuring information transfer, Physical Review letters 85(2), pp. 461–464, (2000)
15. Seth, A. et al.: Granger Causality Analysis in Neuroscience and Neuroimaging, Journal of Neuroscienc(8), pp. 3293–3297, (2015)
16. Yang, D. et al.: Granger Causality for Multivariate Time Series Classification, IEEE International Conference on Big Knowledge (ICBK), pp. 103–110, (2017)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Enabling Self-Diagnosis of Automation Devices through Industrial Analytics

Carlos Paiz Gatica and Alexander Boschmann

Weidmüller Interface GmbH & Co. KG, Klingenbergsstraße 16, 32758 Detmold, Germany
{carlos.paizgatica;alexander.boschmann}@weidmueller.com

Abstract. This paper shows how automation components can be enhanced with self-monitoring capabilities, which are more effective than traditional rule-based methods, by using Industrial Analytics approaches. Two application examples are presented to show how this approach allows the realization of a predictive maintenance strategy, while drastically reducing the realization effort. Furthermore, the benefits of a flexible architecture combining edge- and cloud-computing for the realization of such monitoring system are discussed.

Keywords: Industrial Analytics, Predictive Maintenance, Machine Learning, Edge Computing, Feature Engineering, Self-Monitoring.

1 Motivation and Application Areas

The realization of predictive maintenance strategies in nowadays production facilities is a complex endeavor. Given the rather heterogeneous landscape of typical production facilities, where machines at different stages of their life cycle and from different vendors are combined for a single production line, this situation is even more challenging. In many cases, unplanned downtime is caused by components lacking monitoring capabilities (e.g., dedicated monitoring sensors), which force plant operators to increase the maintenance efforts to guaranty a steady operation. One promising way to drastically reduce the costs of maintenance is the use of Industrial Analytics approaches. Here, the use of data from the production system combined with machine learning methods and domain knowledge leads to the realization of monitoring systems able to automatically detect changes in the behavior of a machine or a component during operation or to predict undesirable situations.

There is a need for flexibility in the realization of Industrial Analytics functions to address the long range of industry applications. For machinery applications, data sets are generated from control systems operating in real time. The applied algorithms need to operate with short reaction times to avoid critical failures or to decrease quality problems resulting on the production of scrap. In these kind of applications, the required sensor data is rather small and the sensor signals are highly correlated to each other. Therefore, an implementation of industrial analytics functions using edge devices alone or in combination with cloud computing brings many advantages, such as short reaction times and decreasing network traffic.

This paper shows the use of Industrial Analytics as means of enabling a condition based- or even a predictive maintenance strategy for simple automation components lacking dedicated monitoring resources. It is shown in section 2 how a flexible architecture combining edge and cloud computation enables the realization of such monitoring system. The process to develop an Industrial Analytics solution is then explained in section 3. Two practical use cases are then presented in section 4, disclosing the potential of this approach to reduce maintenance costs while increasing its effectiveness.

2 Development process of Industrial Analytics solutions

Industrial analytics functions are typically composed of different tasks, as shown in Figure . The figure shows the typical workflow of an industrial analytics application, where data from the different devices are first consolidated in a single data source (*data storage*). The next step is to pre-process the data as preparation for the learning process (*preprocessing*). In this step, relevant features are extracted from the raw data signals, involving the combination of statistical methods with domain-knowledge to select meaningful features.

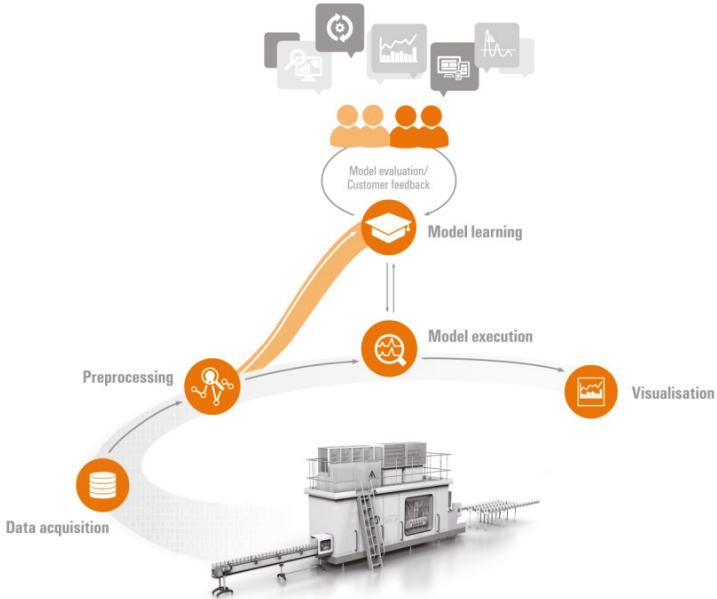


Figure 1: Typical workflow of an industrial analytics system.

The next step is the selection, training and tuning of machine learning algorithms to derive a model from the selected features (*model learning*). Again, the combination of analytics expertise and domain knowledge is key to develop an efficient model. Once developed, the model can be used at runtime to monitor the machine or process (*model execution*). To be useful the results need to be properly visualized (*visualization*). The kind of visualization should be selected according to the role of the person who shall

use this information, e.g., the machine operator, the maintenance manager, etc. The integration of an industrial analytics function in an automation system can be done at different levels, for instance at the machine, or using a cloud platform. These possibilities are explored in the next section.

3 A flexible automation system architecture for Industrial Analytics

In a typical automation system, the continuous stream of heterogeneous data created by machines, actuators and sensors can be used as input for industrial analytic applications such as predictive maintenance. As more and more smart components from the Internet of Things (IoT) domain enter the production facilities, this flood of data will grow dramatically and will become increasingly difficult to manage utilizing a centralized Cloud-based data collection and processing approach. The concept of Edge Computing has recently been proposed to overcome this limitation by providing a distributed computing model where data is processed at the "edge" of a network, i.e., near field devices [SD2016, GJFVR2016].

The core benefit of this approach is to allow for low latency by computing the data where it is created without incurring network latencies, which is essential for real-time condition monitoring applications. Another benefit is scalability: while a traditional centralized approach will no longer be feasible with an increasing number of communicating devices, Edge Computing provides a linear scalability and is needed as augmentation to reduce pressure on network infrastructure. Furthermore, storage and operation cost can be reduced by processing time-sensitive data locally and significantly reducing raw data before being sent to the Cloud. This technique can also be used to preserve privacy by ensuring that sensitive data is pre-processed on-premise so that only privacy-complaint data is transferred to the Cloud. Following the steps from data acquisition to analytics processing and to the visualization of meaningful machine information, various processing steps at different system components are involved. Figure 1 illustrates an example of a flexible automation system architecture implementing Industrial Analytics at Edge-, on-premise- and Cloud levels.

Raw data are acquired by Remote Terminal Units (RTUs) from machines, and process-relevant actuators and sensors over a fieldbus, e.g. PROFIBUS, depicted by green bus connections. An initial pre-processing stage such as filtering can be implemented on these devices. The signals are then collected by a Programmable Logic Controller (PLC) and used to control the system. Additional process-independent components like smart temperature-, vibration- or pressure sensors are typically connected to an Industrial IoT (IIoT) gateway via Bluetooth, WiFi, Ethernet or the emerging 5G [PLZW2015]. These components play an important role in the process of retrofitting and enabling Industrial Analytics services on older machines. Monitoring systems for important control parts that usually don't offer data interfaces by design (i.e. electro-mechanical relays or solenoid valves) can ideally be connected to an IIoT Gateway. We present two practical use cases for these systems in the following section of this paper.

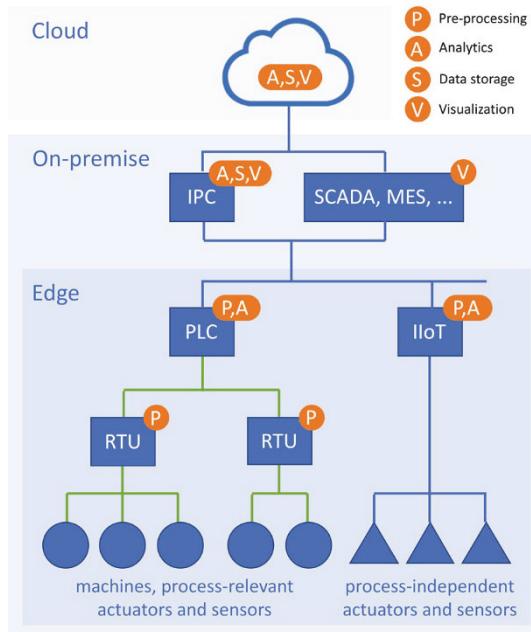


Figure 2: A flexible automation system architecture for Industrial Analytics.

Low latency Edge Analytics functions can be implemented in both, modern PLCs and IIoT gateways. While the PLC can only monitor the devices connected to it, the IIoT gateway typically can access the PLC data in addition to the process-independent component data to generate a larger machine learning model. If necessary, the data density can be further decreased at the Edge level. In addition to data storage and visualization, more complex analytics functions over multiple machines or devices can be performed on-premise by an Industrial PC (IPC) or in the Cloud at the cost of higher latency and increased network traffic. Rich and detailed visualization functions are offered by the Supervisory Control and Data Acquisition (SCADA) or Manufacturing Execution System (MES).

4 Use Cases

In this section two use cases are presented, which show the benefits of enabling simple automation devices with self-monitoring capabilities: Monitoring of electromechanical relays and solenoid valves.

Monitoring of Electromechanical Relays

Electromechanical relays are electrically operated switches that use an electromagnet to mechanically operate a switch to control a circuit by a separate low-power signal.

They are widely used in industrial areas such as plant construction, mechanical engineering or shipbuilding for switching inductive loads, e.g. for controlling solenoid valves.

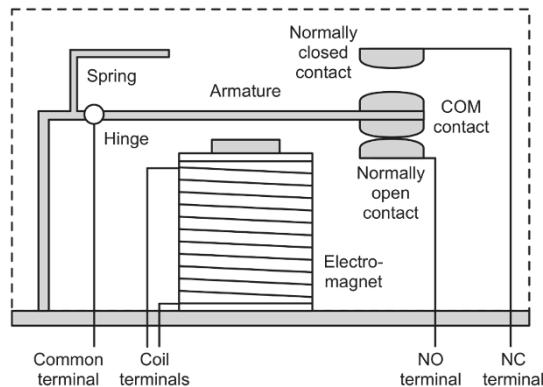


Figure 3: Cross section of a typical electromechanical relay

A simple electromechanical relay consists of an electromagnetic coil, a movable armature and contacts. The armature is attached with a spring so that under normal working conditions it comes back to its original position. If the coil is supplied by the source, a magnetic field causes to attract the armature towards the electromagnet so that the normally open contact (NO) and common terminal contact (COM) connect. This state is shown in Figure 3. When the coil is not supplied by the source, there is no magnetic flux production and the spring draws the armature to its original position so that the normally closed contact (NC) and COM connect. The heavy load on the relay contacts NC and NO that repeatedly occurs while switching inductive DC loads causes premature failure of the relay. Depending on the application, downtime, equipment damage or personal injury can result from component failure. For this reason, it is important to replace damaged relays in time.

In this use case, electromechanical relays were tested for inductive load over their lifetime to develop Industrial Analytics methods for failure detection. In the experimental setup, relays were tested by switching on and off repeatedly under a high DC load. An inductive load was connected to the contact side of the relays, causing an arc between the opening contact surfaces at the moment of switch-off and damaging the relay contacts. This process was repeated until failure of the relay.

A combination of features based on the electric current flow through the relay coil in combination with a Kullback-Leibler divergence-based classifier [KL1951] has been found which allows for a prediction of imminent failure and predictive maintenance. In this study, only features that can be directly measured in the relay without additional sensors were considered. Figure 3 shows an example plot of the classification output.

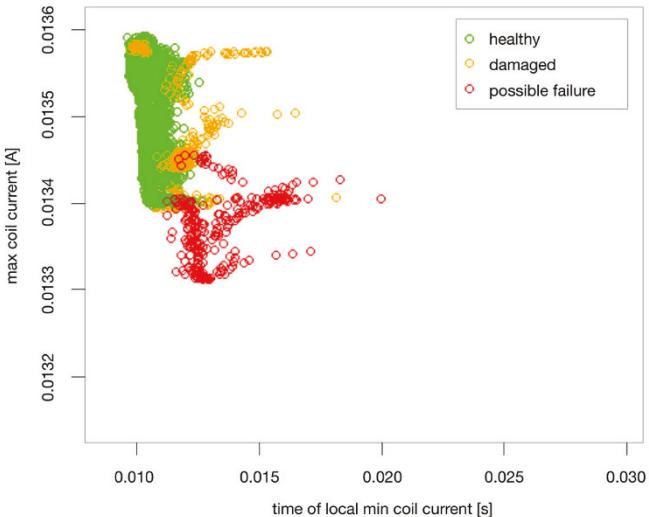


Figure 4: Example classification output of the relay condition monitoring method

Here, the relays were classified into three categories: healthy (green), damaged (orange) and possible failure (red). With the method presented in this paper it is possible to detect an imminent failure due to welding of the relay contacts with high accuracy. In this case, a condition monitoring system can trigger a warning and initiate a predictive maintenance measure before actual damage has occurred. The time remaining in a concrete use case scenario to respond to the imminent failure depends heavily on the switching frequency of the relay being monitored. Based on our experiments, the method presented here allows enough reaction time for applications having high switching frequencies (10 operations per second) or low switching frequencies (1 operation per hour). For this kind of applications, analytics

Monitoring of solenoid valves

Solenoid valves are among the most important control units in today's industry. Especially in the process industry, solenoid valves play an important role because they control the media flow of gases and liquids.

When a current is applied to the magnet winding, the movable magnet armature is attracted, thus releasing the valve plug from the valve seat (see Figure 5). A medium can flow. When switching off the current, the return spring ensures the lowering of the magnet armature and thus the closure of the valve seat by the valve plug. Mechanical loads on the moving parts and the permanent flow of media cause signs of wear inside the solenoid valve. Also, the continuous use under difficult operating conditions, such as high temperatures and vibrating environments, can cause additional wear. Since solenoid valves are often used in safety-critical applications, malfunctions can have catastrophic economic consequences and, above all, put in danger human lives. Not only is wear within a solenoid valve a safety hazard, errors in the signal line (e.g., wire break, short circuit) to the solenoid valve can also cause failures and thus pose a high risk.

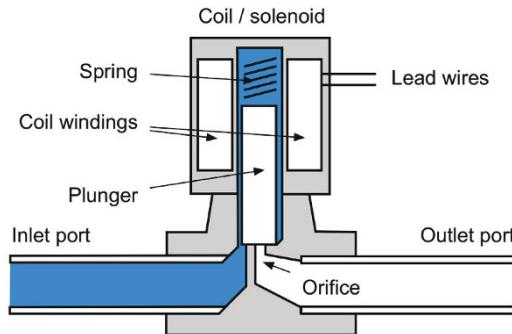


Figure 5: Schematic of a solenoid valve

To prevent premature wear due to wear, the valve or drain in the solenoid valve and the signal line to the valve must be monitored. Four error classes dominate the reports [NRC1987]:

- Foreign matter in the valve (16%)
- Burnt coil / short circuit (15%)
- Worn or defective valve parts (11%)
- Open circuit in coil (9%)

When monitoring solenoid valves, there are two different approaches. The first approach is a rule based approach. During operation, the load current is monitored by means of an electronic component. If the current falls below or exceeds the set limits, the block sends a signal to the controller.

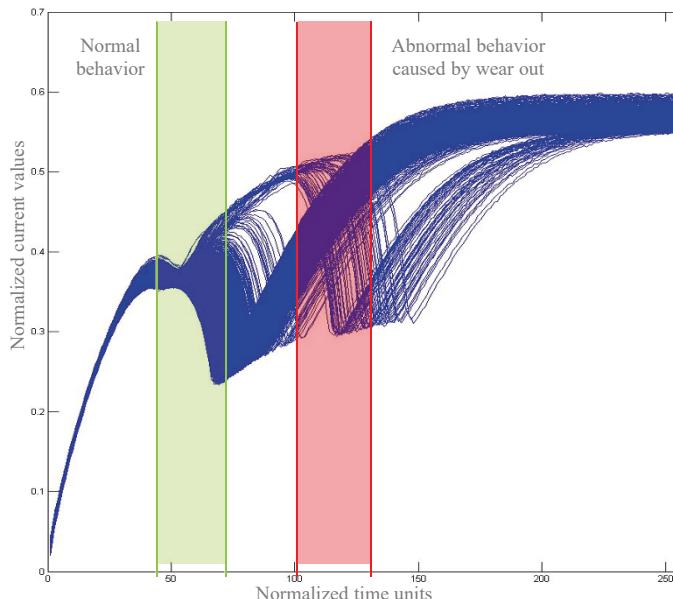


Figure 6: A significant shift in the curves indicates signs of wear on the valve mechanism

With this method, events such as wire breakage, short circuit or overvoltage and undervoltage can be detected and reported. However, changes in the dynamics of the system inside the defined boundaries are not detected.

The second approach pursues the goal of early detection of valve failure. Here, the current waveforms of switching cycles are recorded and compared (see Figure 6). This approach enables device- and application-specific monitoring, because the reference model is created or parameterized during operation. Deviations to a certain extent may indicate a near defect and thus initiate the timely replacement of the valve (see Figure 6). As in the previous case, the realization of this monitoring strategy does not require the use of dedicated sensors, because features extracted from already existing signals are used. This enables the realization of such strategy also for low cost applications.

5 Summary and Conclusions

This paper has shown the use of Industrial Analytics as means of enabling a predictive maintenance strategy. It is shown how a flexible architecture for the realization of data-driven monitoring enables the realization of such monitoring system also for simple automation devices. This is demonstrated by two practical use cases, disclosing the potential of this approach to reduce maintenance costs while increasing its effectiveness.

References

1. [SD2016] W. Shi and S. Dustdar, "The Promise of Edge Computing," in Computer, vol. 49, no. 5, pp. 78-81, May 2016.
2. [GJFVR2016] D. Georgakopoulos, P. P. Jayaraman, M. Fazia, M. Villari and R. Ranjan, "Internet of Things and Edge Cloud Computing Roadmap for Manufacturing," in IEEE Cloud Computing, vol. 3, no. 4, pp. 66-73, July-Aug. 2016.
3. [PLZW2015] M. Peng, Y. Li, Z. Zhao and C. Wang, "System architecture and key technologies for 5G heterogeneous cloud radio access networks," in IEEE Network, vol. 29, no. 2, pp. 6-14, March-April 2015.
4. [KL1951] Kullback, S.; Leibler, R. A. On Information and Sufficiency. Ann. Math. Statist. 22 (1951), no. 1, 79--86.
5. [NRC1987] Aging and service wear of solenoid-operated valves used in safety systems of nuclear power plants: Volume 1, Operating experience and failure identification (Nuclear Regulatory Commission, Washington, DC, USA, 1987)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Making Industrial Analytics work for Factory Automation Applications

Markus Koester

Weidmüller Group, Klingenbergsstr. 16, 32758 Detmold, Germany
markus.koester@weidmueller.com

Abstract. In this contribution, we give an insight in our experiences in the technical and organizational realization of industrial analytics. We address challenges in implementing industrial analytics in real-world applications and discuss aspects to consider when designing a machine learning solution for production. We focus on technical and organizational aspects to make industrial analytics work for real-world applications in factory automation. As an example, we consider a machine learning use case in the area of industry compressors. We discuss the importance of scalability and reusability of data analytics pipelines and present a container-based system architecture.

Keywords: Industrial Analytics, Anomaly Detection, Development Process.

1 Introduction

In factory automation maintainers and operators constantly ask themselves if their assets are operating well or what measures they should take to keep up a good operation and to avoid unforeseen downtimes. Classical condition monitoring approaches, such as signal tracing and threshold mechanisms, only apply for a reactive maintenance scenario, where machine operators usually get informed, when it is already too late to avoid a machine failure. Inspired from recent advance in other areas such as e-commerce and finance, industrial analytics based on machine learning algorithms is gaining attention as a mean to get a deeper insight into the current state of machines or plants. Machine learning is promised to be the key technology to deliver a glimpse into the future of the machine behavior, predicting if and when components are supposed to fail from a statistical point of view under the current operational conditions. In the context of factory automation, machine learning is a relatively new topic, such that the know-how and experience of machinery experts in implementing data analytics pipelines is still limited. Adapting machine learning in the field of factory automation requires not only a sound understanding of the underlying mechanisms of the various algorithms, but also software engineering skills to implement suitable data analytics pipelines for the target machine. Working examples of machine learning implementations at a production level are still rare [1].

This paper gives an overview of the experiences we have gained in creating industrial analytics solutions in the area of machinery and factory automation. The focus of

this paper is more on the challenges in implementing these solutions. Section 2 gives a high-level overview over the functionality of the industrial analytics pipeline, which was considered in the implementations. This pipeline describes the data flow starting from the raw data created by the target machine to the visualization of the analytics results. Section 3 covers the main scope of this contribution by highlighting the challenges, which are a) design considerations to allow for scalability of the solution, b) our underlying process from the first idea of the solution to the final production-ready software, and c) a continuous integration (CI) and continuous delivery (CD) pipeline for automatically building the software solution. In Section 4 we give an overview over an example application, which we have implemented the industrial analytics solution for.

2 Overview of the Industrial Analytics Pipeline

The core concept of the analytics pipeline is present in Fig. 1. Collecting machine data is highly use case dependent and requires to be tailored according to the given data sources and accessibilities of the target machine. To simplify the data processing of the following analytics steps the raw data requires being collected and stored centralized if the target architecture allows. Having a single data source for further data operations of the pipeline, such as a centralized data base, greatly simplifies the data handling.

Preprocessing of the data is a key step to filter out data that has little or even no impact on the modeling success and to create relevant features that represent the actual state of the target machine. As described in the context of data dependencies in [2] the quality of the result of an analytics model greatly depends on the given input features. Besides statistical and data centric approaches, we consider domain knowledge provided by the machine user in the creation of features. Thus we combine expert know-how from the industry application domain and from the data science domain.

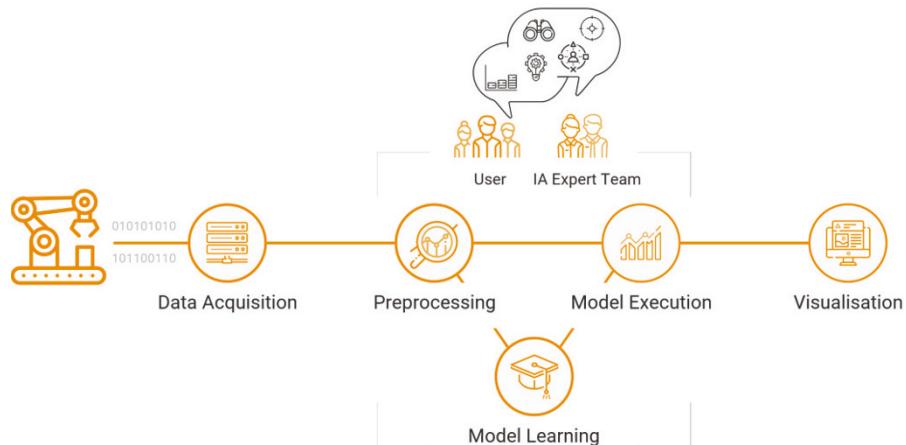


Fig. 1 Industrial Analytics Pipeline

The selected features are used in the two branches model learning and model execution. The selection of the underlying machine learning algorithm highly depends on the target application. Once a model is created it can be used in the model execution branch to compute analytics results. These can be numerical indicators for anomaly detection or contextual information reflecting the current state of the machine. For the scenario of predictive maintenance the output of the model can be e.g. the likelihood of a failure in a given future time interval. This information is finally visualized to support the user in taking decisions for optimizing the efficiency of the machine and for avoiding unplanned down-times.

3 Challenges in implementing Industrial Analytics

In the context of machinery and factory automation industrial analytics is a relatively new topic, where experiences in the technical and organization realization are still rare. In this section we provide an insight into our experiences in implementing industrial analytics in real world applications and discuss aspects to consider when designing a machine learning solution for production.

3.1 Scalability and reusability of data analytics pipelines

In contrast to classical big data application such as natural language processing or image classification, machinery applications typically suffer from little amount of historic data. On the one hand automation technology for collecting machine data at high sampling rates needed for machine learning applications was hardly available. On the other hand there was simply no need to store large amounts of machine data for the given automation application. With the growing awareness of the value of historic data, machine builders and operators start to implement more and more sensor technology to improve the data quality. Thus the amount of data generated by machines will increase in the future as the cost for implementing sensor and storage technology decrease. However, in the current machinery applications, data sets tend to be in the Mega-Byte to Giga-Byte range, allowing for applying small data processing architectures, which should be prepared for scalability to allow for processing larger data sets in the future. To achieve that, we designed a container-based architecture, where the key functions such as the analytics pipeline, frontend user interface, etc. are implemented in separate software containers as shown in Fig. 2.

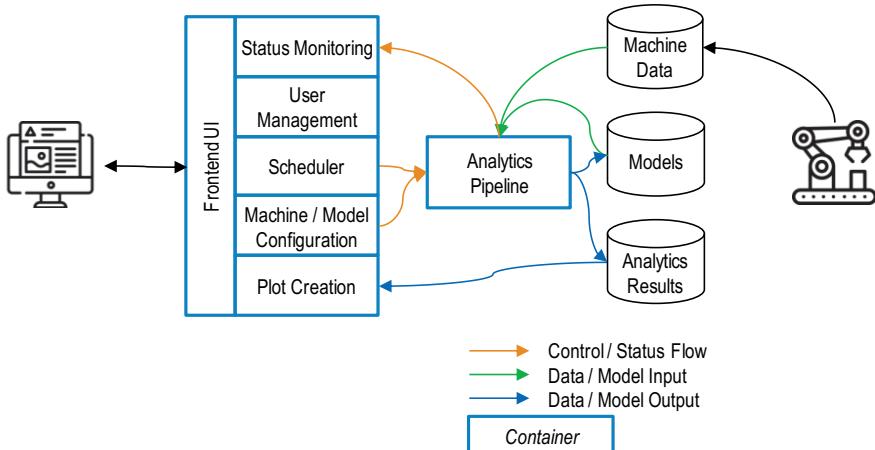


Fig. 2 Container-based Industrial Analytics Architecture

The fronted user interface holds different functionality, which is described in the following: Status monitoring is used to track the state of the analytics pipeline and to inform the user about abnormal behavior. The analytics architecture is designed to handle different users and to provide authentication and user grouping functionalities. Analytics functions, such as model scoring or model learning can be executed in different time intervals, which can be configured in a scheduler. The user can select different models out of the ones given in the model data base and configure and tune the models according to the target machine. The plot creation container is used to generate user-defined plots based on the resulting analytics data.

The machine data is collected and stored in a corresponding data base, which is used as source for the data analytics pipeline container. Besides the machine data the architecture additionally comprises a model data base, where different machine learning models with its pre-processing pipelines are stored.

In a typical flow of the analytics functionality the scheduler triggers the execution of the analytics pipeline, which loads the selected model from the database and applies the model to the specified input machine data. For model scoring the resulting data are written to the analytics result data base, which holds the data for result visualization. For a model learning scenario, the result of the analytics pipeline is a new or updated machine model, which is stored in the model data base, and which can be used for scoring in the future.

The architecture is designed for horizontal scalability and platform independence. Instead of using a single analytics pipeline, the architecture allows for running several analytics pipelines at the same time, which can be used to speed up the execution, or to run different models concurrently. Its container-based implementation allows the architecture to be deployed locally on a single PC (with reasonable amount of available resources) as well as on virtual environments in the cloud.

3.2 Process from idea to production

The industrial analytics solution touches various fields, such as data engineering, machine learning, UI design and systems engineering. Covering the variety of these topics requires an interdisciplinary development team. Typical roles are:

- Application Engineer: They cover the domain knowledge required to get a deep understanding of the target machine application.
- Data Scientist: The data scientist is developing the analytics pipeline from an algorithmic point of view. This involves the feature preparation and selection, as well as the algorithm selection and tuning.
- System Architect: System architects are required to help in defining a layout of the system architecture best suited for the target analytics application.
- Full-Stack SW-Developer: The SW developer implements the selected analytics pipeline to the target system architecture, which can e.g. be an IPC on machine level up to a cloud or hybrid solution.
- UX/UI-Designer: The user interface requires being designed for ease of use providing analytics result information at the right level of detail prepared for the target user group. The UX/UI designer creates the expected UI features and designs the

All project management related tasks are realized by a corresponding project manager. As shown in Fig. 3, we follow a development process, which is inspired by the CRISP-DM process [3]. We tailored the process to meet the special requirements of industrial analytics. Starting from the target definition of the machine learning application, we investigate the quality and quantity of data of the given application and prepare suitable analytics models in the proof of concept phase. In the pilot phase, the model is implemented on the target platform and in the final development phase the missing software features, such as UI functionality and interoperability features are finalized.

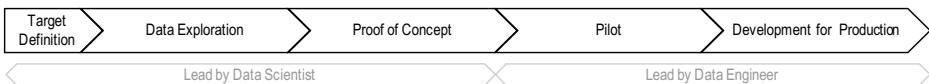


Fig. 3 Development process for data analytics solutions

3.3 CI/CD Pipeline

The development of the proposed container-based analytics solution is realized by a development team consisting of data scientist, data engineers and application engineers. One of the key challenges is to maintain team efficiency in such an interdisciplinary team. A challenge in the development process of an industrial analytics application is the implementation effort for migrating the selected machine learning model from the proof of concept phase to the final software solution in a production environment. A means to reduce this effort is to automate the software build process by continuous integration and continuous delivery (CI/CD). We have implemented a

CI/CD pipeline for the industrial analytics solution and were able to significantly reduce the development effort.

4 Practical Example

To discuss these aspects on a practical example, we consider a real-world machine learning use case in the area of industry compressors. There, we used machine learning algorithms to automatically learn the sensor data distributions of a normal behaving compressor. Consecutively, our models detect deviations from these data distributions and label them as specific anomalies. These anomalies are then predicted by an additional machine learning model to forecast component failures and to prevent unforeseen downtimes.

5 Summary

In this contribution, we focus on technical and organizational aspects to make industrial analytics work for real-world applications in factory automation. As an example, we consider a machine learning use case in the area of industry compressors. We discuss the importance of scalability and reusability of data analytics pipelines and present a container-based system architecture. Furthermore, we share the experience of our development process to bring industrial analytics solutions from idea to production. Based on that process, we present a suitable CI/CD pipeline, which supports our development team to easily bring a machine learning model from the proof of concept phase to production.

References

1. Gatica, C. P.; Koester, M.; Gaukstern, T.; Berlin, E.; Meyer, M.: "An industrial analytics approach to predictive maintenance for machinery applications," 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), Berlin, (2016).
2. Sculley, D.; Holt, G.; Golovin, D. et al.: Hidden technical debt in Machine learning systems. In: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'15), C. Cortes, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 2, pp. 2503-2511. MIT Press, Cambridge, MA, USA (2015).
3. Wirth, Rüdiger: CRISP-DM: Towards a Standard Process Model for Data Mining. In: Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining, pp. 29-39 (2000).

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Application of Reinforcement Learning in Production Planning and Control of Cyber Physical Production Systems

Andreas Kuhnle¹, Gisela Lanza¹

¹ wbk Institute of Production Science, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
andreas.kuhnle@kit.edu

Abstract. Cyber Physical Production Systems (CPPS) provide a huge amount and variety of process and production data. Simultaneously, operational decisions are getting ever more complex due to smaller batch sizes (down to batch size one), a larger product variety and complex processes in production systems. Production engineers struggle to utilize the recorded data to optimize production processes effectively.

In contrast, CPPS promote decentralized decision-making, so-called intelligent agents that are able to gather data (via sensors), process these data, possibly in combination with other information via a connection to and exchange with others, and finally take decisions into action (via actors). Modular and decentralized decision-making systems are thereby able to handle far more complex systems than rigid and static architectures.

This paper discusses possible applications of Machine Learning (ML) algorithms, in particular Reinforcement Learning (RL), and the potentials towards an production planning and control aiming for operational excellence.

Keywords: Production planning and control; Order dispatching; Maintenance management; Artificial intelligence; Reinforcement Learning.

1 Introduction

The productivity of manufacturing systems and thus their economic efficiency depends on the performance of production control mechanisms. Because of an increasing global competition and high customer demands, the optimal use of existing resources is ever more important. Optimizing production control is hence a central issue in the manufacturing industry.

Companies are additionally facing complex manufacturing processes due to high product diversity, lot size reduction and high quality requirements. In the herein considered real-world example of the semiconductor industry, complexity arises through a high number of manufacturing processes and their precision on a nanometer level [1]. Planning and coordinating processes is a challenging task and requires appropriate control methods and decision support systems.

Moreover, production control has to deal with a dynamic and non-deterministic system inside a volatile environment and thus has to handle uncertainty and unexpected incidents [2]. Currently, production planning and control systems such as mathematical programming, heuristics and rule-based approaches are highly centralized and monolithic and not able to meet these needs [3]. Therefore, the dynamic characteristics of production systems are poorly met.

Through the integration of manufacturing components, enhanced process monitoring and data collection, Cyber Physical Production Systems (CPPS) provide real-time data such as order tracking, machine breaks and inventory levels. This makes it possible to apply data-driven techniques, such as Machine Learning (ML) algorithms. Additionally, these are able to adjust to the current system state by analyzing the available data in real-time. This paper shows the successful implementation of a decentral production control system that is based on ML algorithms. The system focuses on the following two use cases: order dispatching and maintenance management. As performance benchmark an existing rule-based heuristic is considered. The real-world use case is taken from a semiconductor manufacturing company that is regarded as a highly suitable example of a cyber physical and digitized production system.

2 Fundamentals and literature review

2.1 Requirements within the semiconductor industry

The semiconductor manufacturing is classically divided into two parts: the front-end, before splitting the wafers, and the subsequent back-end. The front-end comprises all processing steps before cutting the silicon wafer. It consists of several thousand individual processes and lasts between 11 and 20 weeks. Generally, semiconductor manufacturing is considered as one of the most complex manufacturing processes in discrete manufacturing [4]. Between the actual manufacturing processes, control and cleaning processes are required repeatedly. Many of these processes are also performed several times on a wafer so that in general the entire process is not linear. Certain processes are recurrent to build up layers in and on the silicon wafer. Moreover, there are time restrictions between process steps as wafers contaminate quickly when not processed further [1].

2.2 Order dispatching and maintenance management

The assignment of orders to machines for processing is addressed in the so-called order dispatching. Dispatching is an optimization problem that aims to assign orders to resources and hence determines the sequence and schedule of orders. It directly influences the objectives utilization, throughput time (TPT) and work-in-process (WIP).

Next to an optimal order assignment, the robustness of each resource of the system to failures is crucial and has a high influence on these objectives. Therefore, the goal of maintenance management is to maintain availability at minimal cost. Reactive maintenance, i.e. repairs, is balanced with inspection and preventive maintenance measures with the goal to achieve the highest possible uptime of the resources.

Given the challenges of wafer fabrication, order dispatching and maintenance management becomes crucial. Based on real-time process and product data the dispatching and maintenance decisions can be enhanced by ML algorithms in order to optimally match the current manufacturing situation and objectives.

2.3 ML in production planning and control

ML refers to a subsection of artificial intelligence. Many other disciplines of artificial intelligence, such as the processing of natural language or robotics, whose intelligent behavior presupposes a broad knowledge base, are based on this.

There are various industrial applications where ML algorithms are applied with promising results [5]: In [6] an ML algorithm is implemented to control the process parameter power in a laser welding process. The experimental results for a particular setup show that the algorithm generates stable solutions and is suitable for a real-time and dynamic control mechanism. In the context of production control, other authors investigated the usage of ML for order scheduling. The scheduling approaches differ in their overall architecture. The system proposed in [2] and [3], for example, focuses on a highly distributed form, where each resource and each order are considered as intelligent agents. In this kind of architecture resources bid for the allocation of an order depending on the estimated processing cost when being selected. To reduce computational complexity a ML-based solution is presented to estimate the benefit of allocating a job to a specific resource. The implemented ML algorithm uses a table representation in a single objective problem. The work of [7] applies Q-learning to a single-machine scheduling problem and a layout with a few process steps. The order scheduling at each machine and the order release are performed by ML-based agents.

These examples demonstrate the wide range and successful application of ML algorithms in the domain of production engineering. Based on this research the broader application of ML in production planning and control is considered in this paper.

3 Application of reinforcement learning in CPPS

Reinforcement Learning (RL) as one subcategory of ML algorithms addresses the question of how an autonomous, intelligent program (from hereon also named agent) observes and acts in its environment, learning to choose optimal actions in order to achieve a certain goal defined in the beginning. For this, every action of the agent in the environment is rewarded or punished via a scalar number that indicates the desirability of the action, with respect to the overall objectives. The goal of the agent is to maximize this positive feedback [8]. Thereby, the agent explores its environment and learns the optimal connection between the input signal, i.e. the current state of the system, and the action without having to rely on any previous training [9].

3.1 Agent definition

Agents are an essential concept of not only RL but intelligent computing and distributed system design in general [5]. On a functional level, an agent is a computational system that (i) interacts with a dynamic environment, (ii) is able to perform autonomous actions and (iii) acts with regard to a specific objective [5]. To achieve this behavior an agent architecture that has three key components is proposed [10]: For the interaction with its environment the agent needs sensors to perceive relevant aspects of its surrounding and actuators to execute actions. To generate objective-driven actions, a third component, the so-called agent function is required. These characteristics are in line with the general characteristics of CPPS.

In this model, the agent function is the key component for defining the agent's behavior. It determines how the perceived information is processed to decide on actions that lead to a “good” performance with regard to the overall objectives. At the same time, it needs to compromise the agent's experiences. This is crucial to learn the consequences of the agent's decisions. Eventually, the agent function represents a learned model of the environment. The system can consist of several agents with overlapping environments. In that case it is called a multi-agent system [3].

3.2 Reinforcement Learning algorithm

RL applies the ideas of a learning agent-based approach to optimization problems. Because the learning capability is based on repeated interaction with the environment it is often referred to as “trial and error” learning [11]. Despite the existence of many different RL algorithms that vary in the concrete realization of the learning functionality, they follow the same steps in the agent-environment interaction shown in Fig. 1.



Fig. 1. Agent-environment interaction, derived from [11]

The agent perceives the actual state of the environment as a vector S_t . In order to decide on an action A_t the information is processed in the agent function that stores the current policy $\pi_t(a|s) = \mathbb{P}(A_t = a | S_t = s)$. After the action is performed in the environment the agent perceives the new state S_{t+1} and a reward signal R_{t+1} . Note that the environmental transformation is closely linked to the concepts of Markov Decision Processes (MDP). According to the received feedback, the agent adapts its policy. [11]

These steps are repeated in an iterative procedure. As a result, the agent optimizes its behavior in a way to find a policy π maximizing the long-term reward – and therefore a policy that corresponds best to the agent's objectives. [11]

Finding an optimal policy is a iterative process. In each iteration, the current policy π_t is adapted depending on the latest experiences. There are two main techniques to

determine the new policy: (i) value-based and (ii) policy-based approaches. The main difference between both approaches is that value approximation learns the action-value function during the interaction instead of directly learning a policy π . The value function $q_\pi(s,a)$ defines the expected long-term return when choosing an action a in state s following policy π . The policy is then derived from the estimated value of all possible actions in each state. Policy approximation, on the other hand, directly updates the policy function $\pi_t = \pi_t(a|s)$.

Most real-world problems deal with continuous action and state spaces. Storing and updating the policy or value function in a table is therefore computationally inefficient and requires lots of memory space. One possibility is to store the original policy or value function approximatively. Artificial neural networks are widely used for that purpose, as they are capable of approximating complex functional relationships via multiple weights connecting the neurons within the network and allow the adaption of those weights dynamically during the learning process [11]. As a result, neural networks reduce the computational effort by updating a set of weight parameters instead of the values for each state-action pair in each iteration. A dense fully connected feed-forward network is considered in this paper.

Depending on the dimension and the characteristics of the problem, different learning approaches lead to good results. In recent years, new kinds of RL algorithms such as PPO [12], TRPO [13] and DQN [14] were developed to deal with complex problems in different domains. They can be regarded as advanced policy or value approximation algorithms that are optimized with regard to an efficient and stable learning process. The results of this paper are based on these RL algorithms.

4 Case study and experiment results

4.1 Case study setup and description

The considered production system is the production area for wafer implantation. The layout of the production area is illustrated in Fig 2. It consists of three sections with in total eight machines and one entrance and exit lift per section. Regardless of the sections, the machines are grouped according to the principle of job shop production, which can perform the same processing steps. Processing begins with incoming orders at the lifts and the distribution to the respective, pre-defined machines and ends after the order has been processed on the machine and is transported back to the lift. When unloading orders from the lift, access to the first element is always possible. One worker does the transportation between the resources manually. The worker receives the information which order to transport from a central control system. Intermediate storage does not exist, however the machines have a limited buffer in which order batches can be stored before and after processing. The unprocessed batches in the input buffer are automatically fed by the machine according to the FIFO principle and, after complete processing, automatically put into the output buffer.

For this real-world system, a virtual simulation model has been implemented to derive the computational results and evaluate the performance of the RL algorithm. Both,

the simulation model and the RL algorithm, are implemented in Python to be able to implement the bidirectional interaction of the RL agent with the production system.

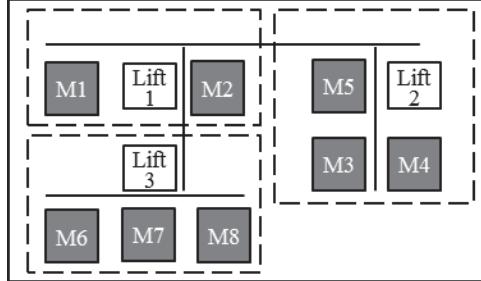


Fig. 2. Layout of production area.

4.2 Intelligent order dispatching

Due to multiple stochastic influences, such as volatile processing times, changing product variants, dysfunctional manufacturing resources and the limited number of transportation resources (just one worker), the system demands a highly flexible order dispatching system. So, the RL-based agent, that decides which order to dispatch next, needs to consider the state of the CPPS in real-time, e.g. the location of all unprocessed and processed batches, tool state information and remaining processing time.

However, it considers just the information that is relevant for the optimal behavior. Just the following state information is taken into account: First, the location of the worker. Second, for each machine one variable for the machine's current availability and the buffer filling state to indicate whether an action ending at a specific machine is possible or not. A second variable based on the existence of a processed order in the machine buffer indicating whether an action starting at a specific machine is possible or not. Two variables for the sum of processing times of unprocessed orders and waiting times of processed orders at each machine. Third, for each entered order one variable for the longest waiting order. A second variable indicates on which machine the longest waiting order has to be processed.

There are three types of possible actions for the agent. Standing at a certain location (machine or lift), the agent can either dispatch an unprocessed order to one out of the eight machines, bring a processed order back to a lift or change its location by moving empty-handed. Additionally, there is the possibility to wait in case there is no order to be dispatched. Moreover, it might be beneficial to wait voluntarily knowing that a batch is available at this location soon.

Objective-driven actions require a feedback from the environment to the agent. This feedback has to be a numeric signal that is transferred to the agent after each action. In this use case a reward of zero is given when the agent decides on an action that cannot be executed by the worker, for example due to machine failure or a buffer overflow. A low value indicates that the agent should avoid such kind of actions, whereas a high value makes the agent behave similarly in the future.

It can be shown that the RL algorithm improves its performance over time, proving that it can be applied as flexible order dispatching control that continuously learns the optimal behavior. Fig. 3 shows the development of the reward signal starting from the initial state where the agent's behavior is completely random. The agent successfully learns a high performance behavior, however not losing the desired flexible behavior. The reward fluctuation points out that the agent is adaptive enough to react to changing conditions of the production system (e.g. disturbances, demand fluctuations). The benchmark FIFO-heuristic approach is based on a set of if-then-rules, e.g. “take the longest waiting batch next” and “first dispatch all batches in one area and move to another area afterwards” (to minimize time consuming area changes). According to Fig. 3 the RL-based algorithm yields a superior performance behavior. After the first iterations the utilization drops to a bottom value. In the end, an overall machine utilization of above 90% is achieved, comparing to a utilization of far below 90% for the heuristic. The same applies for the TPT. Moreover, the heuristic results show an almost stable performance that is not able to adapt to changing conditions. [15]

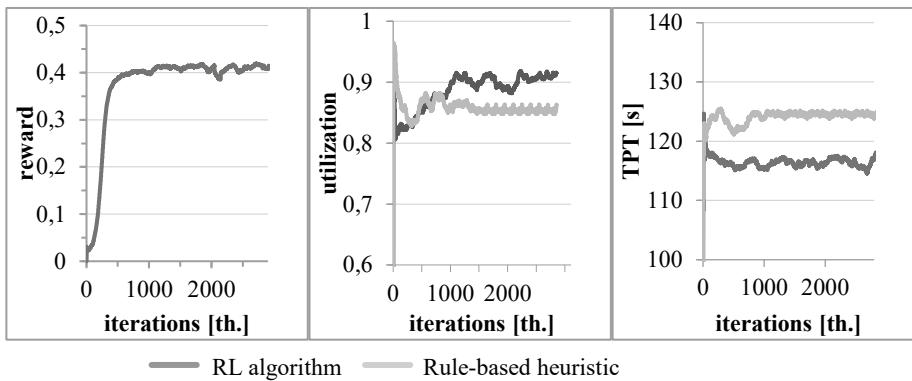


Fig. 3. Reward signal (left), utilization (middle) and throughput time (right); moving average values for 1000 iterations

4.3 Intelligent maintenance management

The aim of the maintenance approach presented in this paper is to predict machine failures and based on this prediction perform the most appropriate maintenance action at the optimal time, which is characterized by a low load of incoming orders, i.e. when the opportunity cost of maintenance are low.

The above presented use case is abstracted and considered as a system that consists of a set of parallel machines, each with a buffer, which receives the orders according to the dispatching. A machine then processes the available orders. The state of each machine is monitored and the state directly affects the performance of the machine, e.g. the operating speed is linked to the achieved output and in case of a failure the machine might only run at a low speed. Initially, the machines operate in a normal mode, where the performance is on the highest level. Each machine fails stochastically. If a critical, failure-initiating value is exceeded, a malfunction begins that ends with the failure after

a certain period. If a machine breaks, a maintenance engineer who is responsible for all machines repairs it and afterwards the machine is back in the desired mode.

In this use case the intelligent maintenance agent is responsible for the decision when and which maintenance action to take. The goal is to reduce the opportunistic maintenance cost, i.e. the optimal action considering the current system load of incoming orders, the cost of the action and the cost of a machine breakdown.

Fig. 4 illustrates the remaining time to failure of a critical state machine at the time the agent performs the action over the learning phase iterations. The agent learns to follow a strategy that brings the action closer to the failure. Additionally, the results prove that the algorithm is able to implicitly learn the prediction and, based on this, perform a suitable preventive action.

Fig. 4 also proves that conducting maintenance as late as possible is able to increase the overall output of the system and comes at lower total cost, since fewer maintenance actions are carried out. The results are compared to two benchmarks: a reactive and a time-base maintenance strategy. The numbers do not take into account the further exploited wear rate of the machine components at the latest possible maintenance time, which is why the actual value tends to be underestimated.

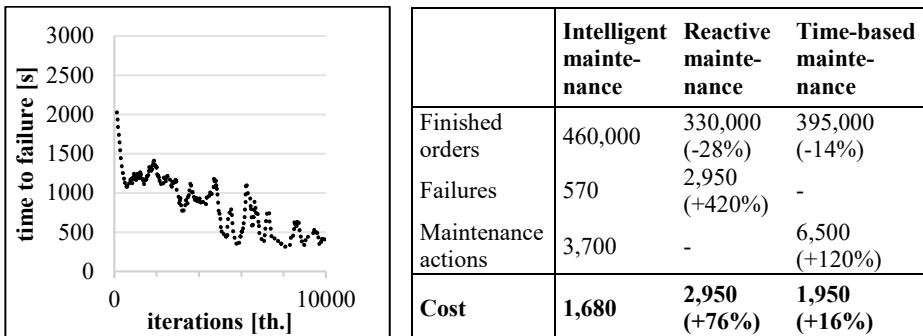


Fig. 4. Remaining time to failure (left, moving average values for 1000 iterations) and cost comparison with benchmark maintenance strategies (right, average values of 40 runs)

5 Conclusion, discussion and outlook

This research has shown that CPPS providing real-time data pave the way for the application of data-driven algorithms to enhance the operational efficiency of production planning and control. RL algorithms are successfully implemented for order dispatching and maintenance management outperforming existing rule-based approaches.

However, ML algorithms are not favorable for all industrial applications. The following properties are advantageous: (i) applications with a limited scope in terms of the number of states and actions (the learning period is dependent on these dimensions), (ii) responsive real-time decision systems (computing the output of a ML algorithm requires just linear operations), (iii) “cheap” training data (the trial-and-error approach is intensively data-driven) and (iv) complex environments that can hardly be described in detail (ability to generalize) [15].

This work brings the application of ML algorithms and the transition towards autonomous production systems one step closer to reality. However, the limitations of ML algorithms and RL in particular still prevail, e.g. in terms of solution robustness. Further research in the area of designing RL algorithms is needed to achieve a broad application also in other areas of production control such as employee allocation and capacity control. Furthermore, research on multi-agent systems is required to broaden the scope of applications.

Acknowledgments

We extend our sincere thanks to the German Federal Ministry of Education and Research (BMBF) for supporting this research project 02P14B161 “Empowerment and Implementation Strategies for Industry 4.0”.

References

1. Mönch L, Fowler JW, Mason SJ (2013) Production planning and control for semiconductor wafer fabrication facilities. Springer, New York
2. Monostori L, Csáji BC, Kádár B (2004) Adaptation and Learning in Distributed Production Control. CIRP Annals 53:349-352
3. Csáji BC, Monostori L, Kádár B (2006) Reinforcement learning in a distributed market-based production control system. Advanced Engineering Informatics 20:279-288
4. Sturm R (2006) Modellbasiertes Verfahren zur Online-Leistungsbewertung von automatisierten Transportsystemen in der Halbleiterfertigung. Jost-Jetter Verlag, Heimsheim
5. Monostori L, Vánča J, Kumara SRT (2006) Agent-Based Systems for Manufacturing. CIRP Annals 55:697-720
6. Günther J, Pilarski PM, Helfrich G, Shen H, Diepold K (2016) Intelligent laser welding through representation, prediction, and control learning. Mechatronics 34:1-11
7. Stegherr F (2000) Reinforcement Learning zur dispositiven Auftragssteuerung in der Varianten-Reihenproduktion. Herbert Utz Verlag, München
8. Thomas MM (1997) Machine Learning. McGraw-Hill, Inc., New York
9. Ertel W. (2011) Introduction to Artificial Intelligence. Springer, London
10. Russel S, Norvig P (2016) Artificial intelligence. Pearson Education Limited, Malaysia
11. Sutton RS, Barto AG (1998) Reinforcement Learning: An Introduction. MIT press, Cambridge
12. Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O (2017) Proximal Policy Optimization Algorithms. arXiv preprint:1707.06347
13. Schulman J, Levine S, Moritz P, Jordan MI, Abbeel P (2015) Trust Region Policy Optimization. International Conference on Machine Learning 2015;1889-1897
14. Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, Riedmiller M (2013) Playing Atari with Deep Reinforcement Learning. arXiv preprint:1312.5602
15. Stricker N, Kuhnle A, Sturm R, Friess S (2018) Reinforcement learning for adaptive order dispatching in the semiconductor industry. CIRP Annals 67;511-514

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





LoRaWan for Smarter Management of Water Network: From metering to data analysis

Jorge Francés-Chust¹, Joaquín Izquierdo² and Idel Montalvo^{3*},

¹Aguas Bixquert, S.L., c/ José Chaix 7, 46800 Xátiva, Valencia, Spain

² FluIng-IMM, Universitat Politècnica de València. Cno. de Vera s/n, 46022 Valencia, Spain;

³ IngeniousWare GmbH Jollystraße 11, 76137 Karlsruhe, Germany;

* Corresponding author. Tel.: +49-162-5459360;
imontalvo@ingeniousware.net

Water distribution systems (WDSs) are large complex infrastructures made from pipes, valves, pumps, tanks and other elements designed and erected to transport water of sufficient quality from water sources to consumers. The amount of the above elements, which can reach up to tens of thousands of links and junctions, their frequently wide spatial dispersion and the WDS characteristic of being very dynamic structures make the management of real WDSs a complex problem [1-4]. However, although the main objective is to supply water in the quantity and quality required, other requirements are essential, namely maintaining conditions far from failure scenarios [5,6], ability to quickly detect sources of contamination intrusion [7,8], minimization of leaks [9-10], etc.

Advances in low powered sensors and data transmission are making their way on the creation of smarter water networks. Despite prices are getting attractive, the return on investment is far from being clear for many water company managers in the water distribution industry. To be prepared to arouse in these managers a real interest in the need for the implementation of an adequate lattice of sensors in their water distribution networks, and to provide them with convincing arguments for their rapid implementation three important questions should be first answered that should be clearly perceived as main support elements in ad hoc decision-making: firstly, how many sensors are needed; secondly, where sensors should be located in order to get the most out of them; and, finally, what to do with the measurements in terms of improving operation and customer services. This contribution addresses the third of these questions without forgetting the other two and present a pilot project at early stage.

There are three aspects crucially important for water utilities and where the correct use of measurements makes the difference on what the company can achieve: reduction of non-revenue water, network operation optimization and provisioning of a quality service. This contribution presents the development of a platform for Smarter Water Network Operation and Management specifically aimed to support the three mentioned aspects. It uses a water network analysis engine to estimate the state of the water network based on measurements taken from the field combined with a mathematical model of the water distribution network. The estimation of the network state is done starting

from the current moment of the analysis and looking 24 hours ahead. This makes possible to optimize the operation of pumps for the next 24 hours considering the price of energy, the expected demands and the available tank capacity in the network. The operation decision of pumps is corrected every hour and can be directly transmitted to the pump station or introduced there by an operator depending on the technology available.

A sensible element in the mathematic modelling of water networks is the estimation of demands. Sub-estimating demands when optimizing the operation of the network can result on a lower quality of the service. Overestimating the demand would result on over costs. The platform developed includes the possibility to receive the consumption measurements directly from water meters installed at the client side or at different interest points of the network. This way, demand values and forecasting algorithms will be periodically getting updated based on the information received. Measuring demand will help in this case not only to improve the results of the operation optimization but also to create a water balance between the water volume supplied and consumed in the network. Water balance is the first analytic step to start estimating non-revenue water in a distribution system. Running water balances for subregions or sectors of the network can help to locate zones with a higher leak impact. Identifying these zones and eliminating their leaks will improve the levels of non-revenue water at utilities. The effect of leaks and as consequence the non-revenue water volume can be also improved by managing properly the pressure of the network based on a robust mathematical model of the distribution system. Additionally, consumption measurements will also help to achieve a better quality in the service: the platform checks the plausibility of consumption and inform both the utility and the client about potential leaks at the client side. Discovering leaks at the client side will avoid the surprise of receiving an expensive invoice with a high consumption due to undetected leaks.

The development of the platform described here is the result of a collaboration between the group Fluing of the Polytechnic University of Valencia, Aguas Bixquert S.L. and Ingeniousware GmbH. This collaboration has resulted in a pilot project developed at a water distribution system managed by the company Aguas Bixquert S.L. For instrumenting the water network, it was considered convenient to use high energy-efficient sensor nodes, preferable battery based and able to communicate across long distance. These characteristics motivated the use of Low-Power Wide Area Networks (LPWAN) [11] technologies for supporting measurements in the pilot project. A LoRaWan [11] antenna was installed at a high point of the zone and it redirects all measurement data to the servers of Ingeniousware where the platform for smarter water network is running. About 30 water meters transmitting consumption via LoRaWan has been already installed at different part of the network. Installation directly at clients will happen in the next phase of the project. A first version of the mathematic model of the water network has been developed and can be visualized directly from the platform. Consumption at all water meters installed can also be visualized as well as transmission statistics. Installed water meters has a temperature sensor integrated and transmit also the temperature value at the installation point. Temperature is a factor that improve significantly the estimation of the water consumption in the network.

The coverage of the data transmission, its stability and the accuracy of the received consumption measurements compared to manual reading of the water meter has been

evaluated. A water meter test bank has been created for these purposes. The most important conclusion of our evaluation is that certification authorities should include an additional error produced at water meters when converting the mechanical movement of the device into a digital signal. Differences from up to 18% were obtained when comparing transmitted values with values read directly from the water meter. It makes think about the necessity of extending the certification of metering devices that consider the maximum error they can have depending on the existing flow. This certification that defines the class of the device and the range of flow where it may work should also consider the potential errors happening when converting the mechanical movement of the water meter into a digital signal. Note that all water meter installed until now in the pilot project are mechanical. A different situation may happen in the case of water meter based on different measurement technology like the ultrasonic but it is still to be tested. At the current stage of the project water meters from only one company has been tested and it is expected to include at least two additional water meters providers for comparison purposes.

References

1. Perelman L. and Ostfeld A. (2012) Water distribution systems simplifications through clustering, Journal of Water Resources Planning and Management Division, ASCE, Vol. 138, No. 3, pp. 218 – 229, [http://dx.doi.org/10.1061/\(ASCE\)WR.1943-5452.0000173](http://dx.doi.org/10.1061/(ASCE)WR.1943-5452.0000173).
2. Izquierdo, J., Montalvo, I., Pérez-García, R., Matías, A., On the Complexities of the Design of Water Distribution Networks. Mathematical Problems in Engineering, Vol. 2012, 1-25, 2012.
3. Ostfeld A. (2012) "Optimal reliable design and operation of water distribution systems through decomposition", Water Resources Research, Vol. 48, W10521, 14.
4. Diao, K., Fu, G., Farmani, R., Guidolin, M. and Butler, D., 2015. Twin-hierarchy decomposition for optimal design of water distribution systems. Journal of Water Resources Planning and Management, 142(5), p.C4015008.
5. Ostfeld A., Oliker N. and Salomons E. (2014). "Multi-objective optimization for least cost design and resiliency of water distribution systems", Journal of Water Resources Planning and Management Division, ASCE, Vol. 140, No. 12, 04014037, [http://dx.doi.org/10.1061/\(ASCE\)WR.1943-5452.0000407](http://dx.doi.org/10.1061/(ASCE)WR.1943-5452.0000407)
6. Herrera, M., Abraham, E., & Stoianov, I. (2016). A graph-theoretic framework for assessing the resilience of sectorised water distribution networks. Water Resources Management, 30(5), 1685-1699.
7. Islam N, Farahat A, Al-Zahrani MAM, Rodriguez MJ, Sadiq R (2015) Contaminant intrusion in water distribution networks: review and proposal of an integrated model for decision making. Environ Rev 23(3):337–352
8. Nafi, A., Crastes, E., Sadiq, R. et al. Intentional contamination of water distribution networks: developing indicators for sensitivity and vulnerability assessments Stoch Environ Res Risk Assess (2018) 32: 527. <https://doi.org/10.1007/s00477-017-1415-y>
9. Covas, D. and Ramos, H., 1999. Practical methods for leakage control, detection, and location in pressurized systems. In BHR Group Conference Series Publication, Bury St. Edmunds; Professional Engineering Publishing, Vol. 37, pp. 135-152.

10. Candelieri, A., Conti, D. and Archetti, F., 2014. A graph-based analysis of leak localization in urban water networks. Procedia Engineering, 70, pp. 228-237.
11. Hhutsoane O., Isong, B., Abu-Mahfouz, A., 2017. IoT devices and applications based on LoRa/LoRaWAN. IECON 2017 – 43rd Annual Conference of the IEEE Industrial Electronics Society.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

