**REFLEXION**

*React to Effects Fast by Learning, Evaluation, and eXtracted InformatiON*



**Work Package 1 – State of the Art**

D1.3

© REFLEXION consortium

Axini – Barco – Océ Technologies – Philips – Siemens ISW – Synerscope – TNO – Yazzoom

**Author(s) :** Siemens ISW, TNO-ESI

**Version, Date :** v5.0, 24/07/2018

**Confidentiality :** public

# Abstract

This deliverable serves to establish the current industrial practices for each OEM and the leading machine learning and data analytics methods to be applied by the SMEs. As developers for modelling and model-based-tests, Axini investigates how to improve test design using in-service data. Barco aims to characterise how its products are used by customers using log data events to improve designs of future product generations quickly. Océ outlines its typical log data production, fault diagnosis procedure and a range of possible errors one may find in the log data. Philips IGT summarizes its current system and user profiling methodology which it aims to improve using data analytics. Philips MRI focuses on fault diagnostics, which is typically performed manually by a service engineer. With the large amount of data collected on the MRI machines, faults may be able to be detected before machine downtime occurs. Siemens ISW focuses on first principles and numerical modelling of defective mechanical components that are commonly found in automotive applications and prone to failure. Next, an overview of machine learning algorithms to mechanical applications is presented. Synerscope provides an overview of modern visualization techniques for high dimensional data and details the challenges associated with different types of data. Finally, Yazzoom presents the current techniques in anomaly detection in machine learning and details different methods and how they applied to log based and time series data. Finally, commonalities are identified.

# Change log

|  |  |
| --- | --- |
| Version, Date | Description |
| v0.1, 25/04/2016 | First draft by Cameron Sobie (WP1 leader) |
| v0.2, 01/08/2016 | Ready for internal review |
| v0.3, 06/08/2016 | Reviewed by TNO (Bas Huijbrechts) |
| v1.0, 17/08/2016 | Rework and version 1.0 release by Cameron Sobie (WP1 leader) |
| V1.1, 13/06/2017 | Commonalities added by Emile van Gerwen (TNO) |
| V1.2, 18/07/2017 | Reviewed by SISW (Cameron Sobie) |
| V2.0, 30/07/2017 | Layout updates by Emile van Gerwen (TNO) |
| V3.0, 18/06/2018 | Expansion to include Lessons Learned |
| V4.0, 23/07/2018 | Full version of document ready for internal review |
| V5.0, 24/07/2018 | Full version, ready for release |

Table of Contents

[Abstract 2](#_Toc520054901)

[Change log 2](#_Toc520054902)

[1. Introduction 6](#_Toc520054903)

[2. Axini 6](#_Toc520054904)

[2.1. State of the Art 6](#_Toc520054905)

[2.2. Lessons Learned 6](#_Toc520054906)

[3. Barco 7](#_Toc520054907)

[3.1. State of the Art 7](#_Toc520054908)

[Existing State-Of-The-Art with respect to data collection, data transport & storage and data analytics 7](#_Toc520054909)

[Improving system-wide root-cause & trend analysis 7](#_Toc520054910)

[Development of improved predictive models for End-of-life determination for a component 8](#_Toc520054911)

[Develop new data-driven services and revenue streams enabled by data analytics 8](#_Toc520054912)

[Differences between INVALUE and REFLEXION 8](#_Toc520054913)

[3.2. Lessons Learned 9](#_Toc520054914)

[Collection and augmentation of data 10](#_Toc520054915)

[Exploration and visualization of data 11](#_Toc520054916)

[Anomaly detection 13](#_Toc520054917)

[Understanding the value of data 14](#_Toc520054918)

[4. Océ 15](#_Toc520054919)

[4.1. State of the Art 15](#_Toc520054920)

[Error detection handling and recovery: MRE 16](#_Toc520054921)

[Error detection handling and recovery: ORE 16](#_Toc520054922)

[Runtime contradiction 16](#_Toc520054923)

[Error detection: Warning 16](#_Toc520054924)

[Error detection handling and recovery: PE 16](#_Toc520054925)

[Counters (preventive maintenance) 16](#_Toc520054926)

[Logging 16](#_Toc520054927)

[Input, Output, Special, System test 17](#_Toc520054928)

[Adjustment test, automatic process optimization 17](#_Toc520054929)

[Handicapping 18](#_Toc520054930)

[4.2. Lessons Learned 18](#_Toc520054931)

[5. Philips IGT 19](#_Toc520054932)

[5.1. State of the Art 19](#_Toc520054933)

[From X-ray system to Intervention solution 19](#_Toc520054934)

[User profiling in system test 19](#_Toc520054935)

[State-of-the-art of user profiling 20](#_Toc520054936)

[5.2. Lessons Learned 20](#_Toc520054937)

[Data processing and providing insights 20](#_Toc520054938)

[Hierarchical Usage Profiling and Model-Based Testing 22](#_Toc520054939)

[Usage Profiling and Machine learning 22](#_Toc520054940)

[Connecting the dots 23](#_Toc520054941)

[6. Philips MRI 24](#_Toc520054942)

[6.1. State of the Art 24](#_Toc520054943)

[Way of working 24](#_Toc520054944)

[Features 25](#_Toc520054945)

[Delta with vision 25](#_Toc520054946)

[6.2. Lessons Learned 25](#_Toc520054947)

[7. Siemens ISW 27](#_Toc520054948)

[7.1. State of the Art 27](#_Toc520054949)

[Condition monitoring in mechanical industries 28](#_Toc520054950)

[Modelling Component Failures 29](#_Toc520054951)

[Machine Learning Applications to Mechanical Components 33](#_Toc520054952)

[7.2. Lessons Learned 35](#_Toc520054953)

[Simulation-driven Machine Learning 35](#_Toc520054954)

[Machine Learning in an Engineering Context 37](#_Toc520054955)

[Fleet-based Condition Monitoring 39](#_Toc520054956)

[8. Synerscope 41](#_Toc520054957)

[8.1. State of the Art 41](#_Toc520054958)

[Introduction 41](#_Toc520054959)

[Characterization of the visualization problem 42](#_Toc520054960)

[Visualization techniques for time-oriented data 44](#_Toc520054961)

[8.2. Lessons Learned 45](#_Toc520054962)

[Visualization component improvements 45](#_Toc520054963)

[Event streams 47](#_Toc520054964)

[9. Yazzoom 48](#_Toc520054965)

[9.1. State of the Art 48](#_Toc520054966)

[Anomaly detection in machine learning 49](#_Toc520054967)

[Anomaly detection on time series data 51](#_Toc520054968)

[Anomaly detection for log analysis 51](#_Toc520054969)

[9.2. Lessons Learned 52](#_Toc520054970)

[10. Commonalities 53](#_Toc520054971)

[10.1. Data collection and storage 53](#_Toc520054972)

[10.2. Data content 53](#_Toc520054973)

[10.3. Data context 54](#_Toc520054974)

[10.4. Data quantity 54](#_Toc520054975)

[10.5. Data versioning 54](#_Toc520054976)

[10.6. Data usage 54](#_Toc520054977)

[10.7. Dashboards 54](#_Toc520054978)

[11. References & citations 55](#_Toc520054979)

1. Introduction

This document contains a summary of the state of the art or current practices for each OEM and SME within the project. Barco, Océ and Philips focus on what is captured in log based data, how it is currently processed and its current uses and exploitation, whereas Siemens ISW (SISW) provides a summary of the leading research on physical defects and their modelling and related applications of machine learning. Each SME provides a summary of modern techniques in model-based-testing, visualization, and anomaly detection from Axini, Synerscope, and Yazzoom, respectively.

1. Axini
   1. State of the Art

Axini is the developer of a powerful tool-set for modelling and model-based-testing. Our toolset is based on the IOCO test-theory of Jan Tretmans [1]. It has a formal underpinning with process algebraic semantics and labelled transition systems.

Axini expanded the theory with a pragmatic modelling language that supports generic data types and time. We invented several test-derivation strategies and techniques in order to achieve optimal test coverage.

There is a long history in Model Based Testing (MBT) research. One of the first papers is already from 1956 of Moore [2] Although already some years old, a nice overview of model-based testing can be found in [3].

Model-based testing is a form of model-driven development applied to testing. State of the art is that MBT automates all the steps in the test-process: test-case creation, test-case execution, and checking the outcome of the execution

REFLEXION advances the state of the art in that it looks at new ways to apply the knowledge in the field for test-case derivation. It wants to include feedback from the field in the development process. For MBT this influences modelling and test-derivation. Therefore, we hope to improve the theory and tools for modelling and test-derivation.

* 1. Lessons Learned

We at Axini make models of systems from a design perspective. The model describes the behaviour of the system. It contains the system's components and the interfaces through which they interact. These models can become the specification and replace the design documents. They are checked for correctness before the system is implemented and are used to test the system in every iteration of its development.

Models can also be created from a test perspective. These are test models. They do not describe the behaviour of the system and thus cannot become a formal specification of the system. Their use is limited. They are only useful for generating test cases and testing the system. We rarely make test models.

When we started working on REFLEXION, in particular on Philips IGT case study and on the Allura Xper X-ray systems, we did not have access to design documents. The only information available of the system was its production logs. Philips already has a way to replay traces from a log against development systems using their Test automation Framework (TAF) but that is not sufficient. Philips likes to increase the coverage and they like to automate the generation of test-cases (at the moment their test engineers have to manually create test scripts to test the system). Philips also like to use their test systems more optimally.

Considering the absence of design documents, we began to create a model of the user of the system (a test model). We also implemented test strategies for user profile testing (Our contribution is available in detail in REFLEXION deliverable D4.1). We were surprised how quickly we were able to generate executable test cases based on a user profile. We were able to expand the model very quickly. We learned that with user profiles we can model less and abstract from details that are otherwise necessary for a design model. We also learned that through user profile driven model based testing we can go beyond merely replaying traces from production against development systems. We can test the system with far more data variations and interactions.

1. Barco
   1. State of the Art

### Existing State-Of-The-Art with respect to data collection, data transport & storage and data analytics

Barco provides devices for a wide range of markets, including digital cinema projectors and medical imaging devices. Visualization solutions for these markets typically have high RAS[[1]](#footnote-1) requirements. Better control over – and limiting the frequency of – planned (maintenance) and unplanned (break-downs) interruptions are considered important differentiators for these products.

Existing device management platforms for digital cinema projectors and medical imaging devices – respectively CineCare and QAWeb – that allow inspecting the devices in the field have, or are, being augmented within the scope of the INVALUE project (ITEA Project #13015) so that they allow proactive preventive maintenance instead of the now commonplace reactive corrective maintenance. These activities are focused around advanced physical system and component analysis and are backed by strong use cases:

### Improving system-wide root-cause & trend analysis

New and improved means to perform system-wide root cause & trend analysis have been developed to facilitate the paradigm shift from reactive curative corrective repair actions and schedule based maintenance actions towards proactive corrective repair actions with a specific focus on reducing the *wear-out costs* as a product ages:

An aggregate of data mining components have been developed, evaluated and interconnected to allow to:

* Interactively query all the available data about a certain device, regardless of the information source the data comes from.
* Analyse the evolution of certain data points in time (i.e. time series analysis, outlier detection, clustering, …)
* Detect significant markers in the current date-streams (trends, correlations …) that herald forthcoming product failures.
* Better understanding of real-life usage of our product base in the field.
* Identify and isolate those aspects or interactions in a product that are significant contributors to premature warranty costs and reflect this back to product designs.
* Let different stakeholders explore the data in an intuitive, graphical and efficient way by providing appropriate user interfaces or API’s and defines the necessary infrastructure around it to handle concurrent data exploration requests.

### Development of improved predictive models for End-of-life determination for a component

Barco gained a lot of knowledge containing a) the correct set of (physical) data points, their logging frequency and the data granularity and b) the necessary machine learning techniques to bridge the discrepancy between predicted MTBF[[2]](#footnote-2) values and observed MTBF values of physical components and products in the field. This is both the case for cinema projector lamps and healthcare display backlights where we can now not only more accurately predict their aging behaviour, leading to lower accrued warranty costs, but even can quite accurately predict the point in time where an individual lamp will fail or a single backlight will start to operate outside of its regulatory envelope (brightness too low).

### Develop new data-driven services and revenue streams enabled by data analytics

Till recently, the focus of Barco was on selling products and standard services (i.e. warranty). It had no knowledge nor framework installed to sell additional services based on the insight that deep data analytics on the full range of data sources can bring. This is currently in the process of being changed: new invoice models (for example “light-lease”[[3]](#footnote-3) for projector lamps) are being tried out and the associated ecosystem put in place (supply chain, sales, inventory…)

### Differences between INVALUE and REFLEXION

In the table below, a comparison is made between the INVALUE scope and the REFLEXION challenge to highlight the different challenges targeted by both projects.

|  |  |
| --- | --- |
| **INVALUE scope** | **REFLEXION challenge** |
| Data-points logged relate mostly to physical quantities (voltages, currents, temperatures, light output …) that tell something on the health, operational envelope and risk for imminent failures of the physical hardware components themselves as they age. | Data-points to be logged are mostly operational parameters that do not directly reflect the health status of physical components. The data-points mainly reflect the current status, API-calls and state transitions in software & networking stacks in the different Barco devices (encoders, decoders, network management software). |
| The networking infrastructure is only used as a means to transport the logged artefacts from individual projectors or healthcare displays to a central data store. | The dynamic and distributed communication (i.e. video-streaming multicast over virtual LANs) happening over the networking infrastructure in a hospital (i.e. multiple interconnected operation rooms) is itself a main target for data-acquisition and data analytics. |
| There is a clear focus on accurate end-of-life prediction of individual physical, opto-electronic components (cinema projector lamp, display backlight …) or in general, wear-out and aging trends (root-cause-analysis).  The implication is that both a) the set of data-points to be logged and b) the kind of machine learning techniques can be tuned towards the goals set at the start of the project. | A key goal of the REFLEXION project is to learn how a distributed collection of Barco networking devices (encoders, decoders, …), that are incorporated in a larger networking infrastructure (outside the control of Barco) are effectively used in the field by our customers and how robust this federated infrastructure respond to dynamic changes in these environments. As such, it is not clear which set of data-points are relevant and which kind of data-analytics and visualisation techniques will prove most beneficial. |
| Timing or (timing accuracy) at which each data-point is logged and strict timing correlation between the different data-points is not important. | Correctly inferring the causal and temporal order of events that happen in the different distributed devices in a network without the availability of a (sufficiently accurate) global clock is of paramount importance. |
| Inferring how physical components effectively behave and age in our customer environments with a quick feedback loop to product development and customer service are key objectives in this project. | Inferring early in the product deployment phase how our products are effectively used by our customers (instead of how we assume that they will be used) with a quick feedback loop to product development is a key objective in this project. |

* 1. Lessons Learned

Modelling normal behaviour and supporting root-cause analysis of the Nexxis distributed video management system is key to Barco NV in the REFLEXION project. For this purpose, Barco has been analysing log data coming from both internal validation setups and hospitals.

In the course of the project, Barco has learned in all areas of a typical data science workflow: in the areas of collection and augmentation of data, exploration & visualization of data, anomaly detection and the understanding of the value of data.

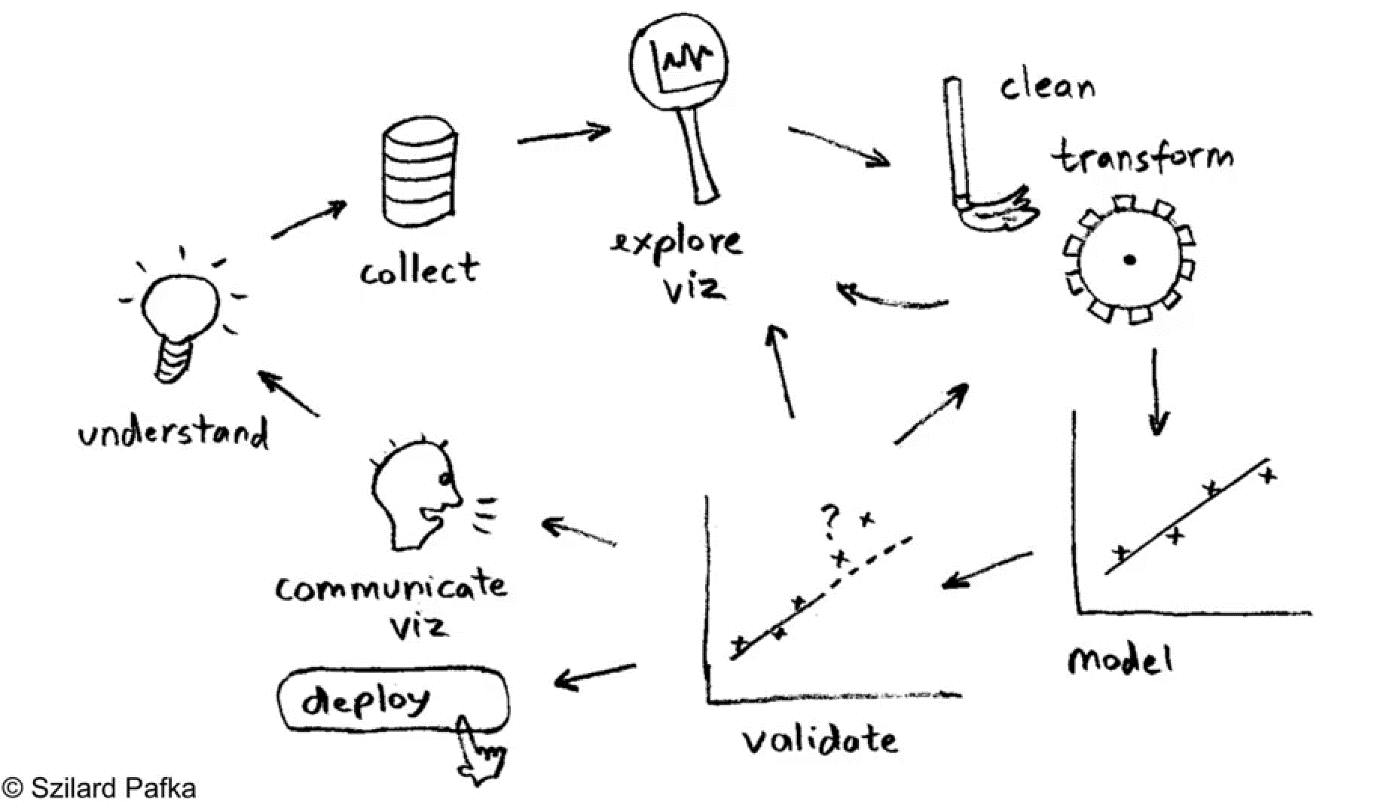


Figure 1. Data science workflow in REFLEXION

### Collection and augmentation of data

Barco has explored various ways of data collection: through the existing tunnel from the system integrator to the hospital or through a new tunnel from the hospital to Barco Nexxis Cloud Services. With the customers that have been solicited until now, customers clearly favoured that Barco collected the system log data through the existing tunnels from the system integration to the hospital. Next to the technicalities, reusing the existing tunnel was favoured as all legal contracts are in place in the existing eco-system of system integrator and hospital.

Barco learned to improve on the log quality. A log server was introduced that gathers log data in real time to overcome the timestamp problems of some embedded devices. The timestamp formats were all set to the ISO 8601 to easier detect when a NTP synchronization issue occurred.

To make logs more readable, a stream of human readable log events was introduced. This is of great value to our support services and the system integrators, to unravel what has happened when an incident is reported.

During the course of the REFLEXION project, it became very clear that the existing log data does not suffice for root cause analysis. Barco has learned to do a data analysis with the different stakeholders such as R&D, product management and service organisation of both Barco and the the system integrators. Within the data analysis, all data needs were tabled out in a comprehensive way and new needs were identified.

As a consequence of the above data analysis, several new metrics were introduced next to log data, both on a NMS[[4]](#footnote-4) level (registry size, buffer size, system metrics), encoder/decoder level (CPU load, RAM used, …) and on network switch level (packets lost, bandwidth used, …). As R&D faced field issues to troubleshoot and correct problems, R&D indicated that these metrics are instrumental in detecting anomalies in the usage and correct functioning of the system.

### Exploration and visualization of data

Goal of the project is to enable the service organization and R&D of both Barco and the system integrators to analyse field issues. Therefore, many ways of data exploration and visualization tools have been explored to support the workflow of service engineers doing root cause analysis. In the course of REFLEXION, Barco has learned that a single visualization tool does not suffice in order to explore data and do root cause analysis. A dashboard has been constructed that allows the service engineers to efficiently manipulate, filter and visualize data from experiments and log streams.

The heatmap visualization (cf. Figure 2) allows the service engineer to oversee activity on a network level. The color coding is adapted on a device level to indicate a certain degree of usage (by calculating certain quantile values as thresholds), reflected by number of messages per minute.

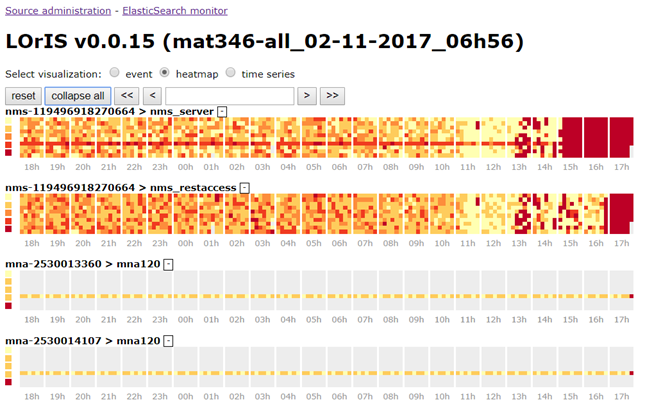


Figure 2: Heatmap of log activity on Nexxis setup

The timeline visualization (cf. Figure 3) plots the log lines of a query in an intuitive fashion and allows for exploration (zoom in/out and pan), with a separate lane for each device:

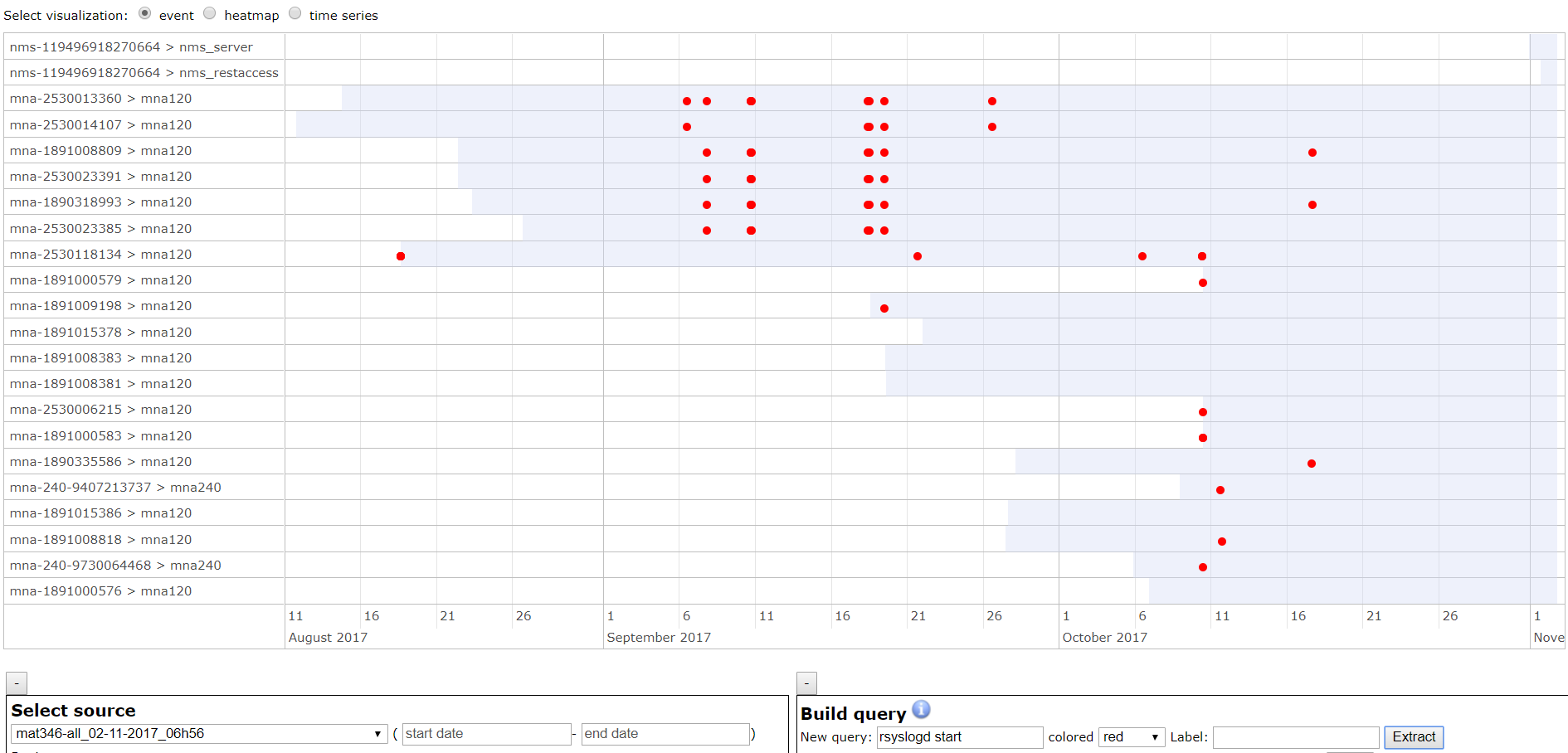


Figure 3. Timeline Vizualization of log data

A scatterplot visualization (cf. Figure 4) allows to visualize different sources. In the below example with a collection of 130 logfiles, a single source can be singled out. The bubble size indicates the relative size of the selected anomaly score for the given observation. A histogram gives an indication of the distribution of the anomaly score that was selected.

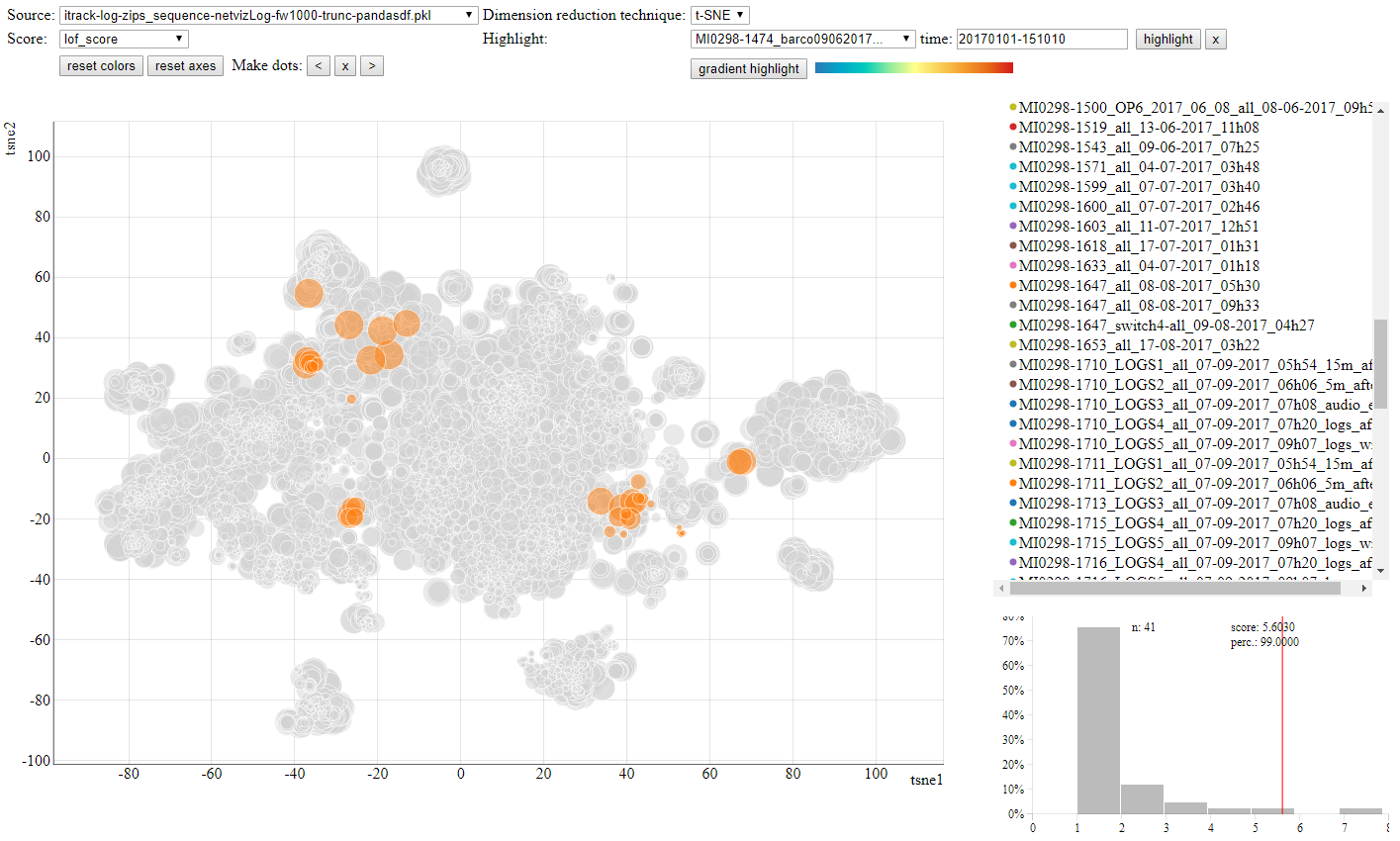


Figure 4: Anomaly scatterplot with highlighting of specific data source and histogram of anomaly scores

Because we’re dealing with time sensitive data, a “gradient highlight” button was introduced to highlight the dots in a rainbow color scheme, which gives a nice feel for how the observations evolve through time. Additionally, it’s possible to highlight a specific point in time for the selected data source (cf. Figure 5).

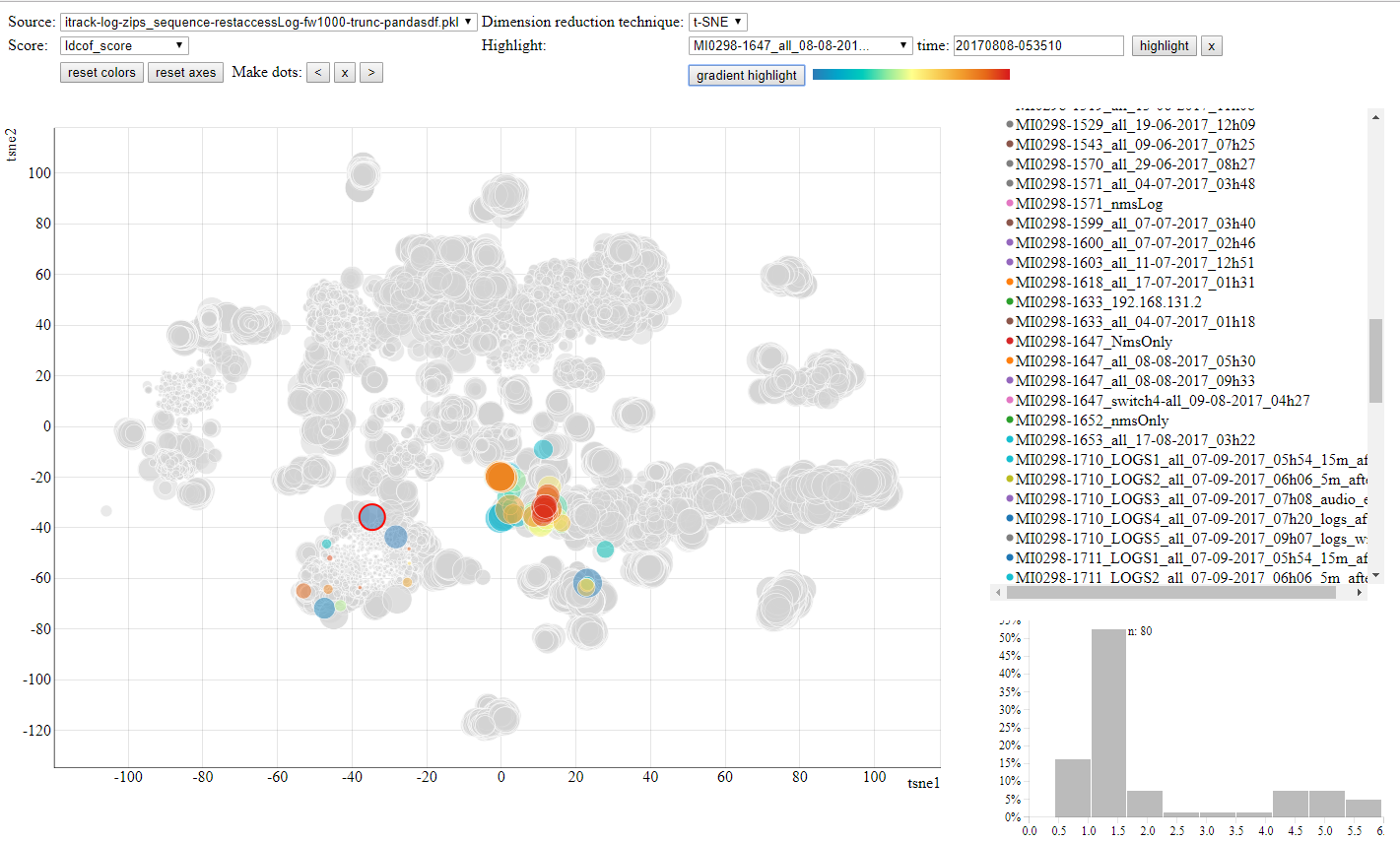


Figure 5: Gradient highlight on anomaly scatterplot

The metrics that are gathered from the devices are visualized as both time series and density plots, such that a support technician could easily evaluate whether a given value is an outlier or not (cf. Figure 6).

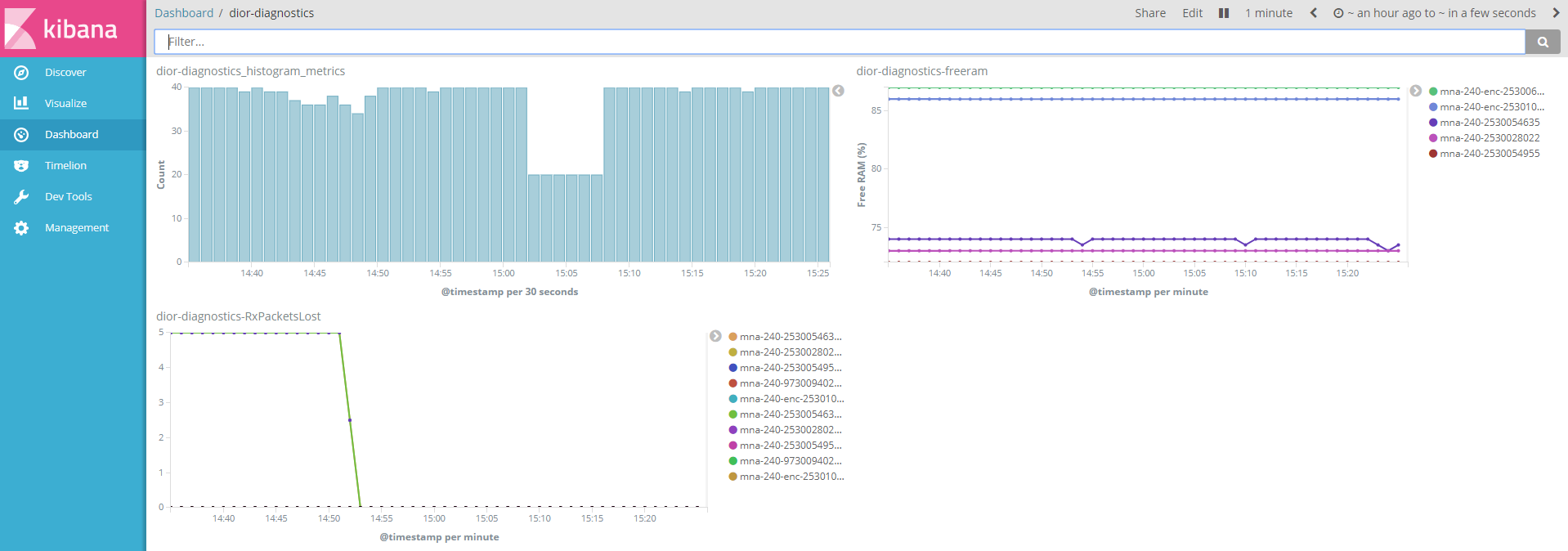


Figure 6. Timeseries and Density Vizualization

### Anomaly detection

Regarding anomaly detection, a few notable learnings are in place. Firstly, there is often not a clear definition of an anomalous state and “labeled” data is hard to come by. In text/log analysis, “unexpected behavior” in itself is very vague and although certain unsupervised techniques exist, they often don’t provide actual insights into the nature of the issue. As interpretability of the results is a key requirement if we wish to report anything to the user, these methods are not to be considered.

Different methods allow for outlier analysis, and vary in terms of complexity. Consider a multivariate dataset with device metrics such as CPU load, free RAM and fan speed. A lot can be said about “normal” behavior when analyzing their univariate distributions, and correlation analysis and scatterplots can also uncover relations between parameters.

Another key learning is the unbalanced class problem: detection of a specific issue is often so rare that the ratio of anomalous/erroneous data samples is completely out of balance. Training models to classify this type of problem often results in a system that doesn’t report any errors or contains a lot of false positives. Lastly, the analysis of time series data rarely is as straightforward as textbook data analysis problems suggest. How to deal with missing or qualitatively lacking data problems can be of great impact to the accuracy of the solution.

Another important factor in evaluating anomaly detection algorithms is how well these scale with large data quantities. When scaling is inadequately possible, real-time analysis/inference is not possible.

A number of univariate and multivariate unsupervised anomaly detection algorithms have been investigated. Some examples are Linked Outlier Factor, Local Outlier Probability, Cluster-Based Local Outlier Factor, Local Density Cluster-Based Outlier Factor, One-class SVM. These algorithms do not consider sequences of time series data.

Another approach in anomaly detection is to evaluate the likelihood of a log sequence. Several deep learning architectures are evaluated for this purpose, such as a sequence to sequence (seq2seq) neural net and an autoencoder.

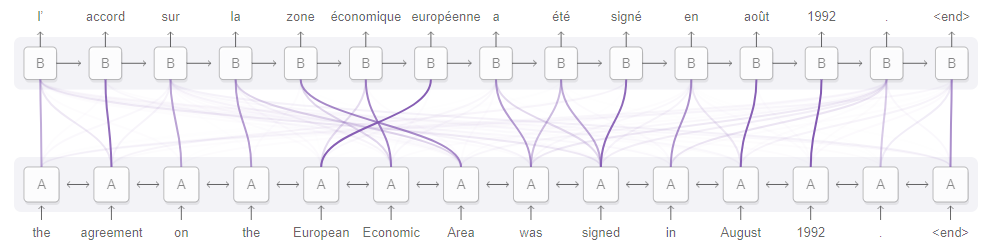


Figure 7: seq2seq model with attention

Barco is still evaluating which algorithms with what hyperparameters contribute to a viable solution using the log data from validation setups and customers, Having more context to the data would have helped and add more value to the customer.

### Understanding the value of data

The REFLEXION project brought with it huge challenges. Due to low data quality and availability, no systematic data gathering effort took place before the advent of REFLEXION. Thanks to the REFLEXION project, a big change has taking place and the awareness of the great potential of data as a catalyst to provide product usage insights is growing amongst the various stakeholders in the product team.

The results of the REFLEXION project are spreading throughout the Barco organisation to the other divisions very rapidly. These divisions are considering these results.

Interpretability is another key challenge. Metadata and data annotation allow us to put results of anomaly detection algorithms in perspective. If we have extensive contextual information, it can help to know what the difference is between several log sequences, or why a certain metric like CPU load is so high in log scenario 1 versus scenario 2. Contextual awareness and (meta)data availability are definitely very important challenges to overcome to succeed in an analytics project.

1. Océ
   1. State of the Art

This section focuses on the technologies traditionally implemented in embedded software that support diagnosis, traditionally defined as the analysis and fixing needed to get from a defective to a working printing/scanning system. The scope is wider than diagnostics during the use phase at the customer since diagnostics also plays an important role during designing, production and recycling of a printing system.

The main requirement for the diagnosis solution is that it is able to indicate the cause of a failure. The requirements have a strong dependency with the life phase of the engine and the involved actors. The requirements for a diagnostic solution used by the helpdesk in the use phase are very different from those stated by an engineer in the design phase. (An engineer can use log files for the analysis, where the helpdesk relies on error codes that the customer reports.)

There are three main maintenance strategies service can choose from. Most commonly used is *corrective maintenance* where no action is undertaken until the print system is defective. The requirements to the diagnosis solution are that the failure should be detected, the resulting damage should be minimized and the defective part should be indicated.

In case of *preventive maintenance,* a part will be replaced when it has experienced a certain number of load cycles. The threshold is determined by the lifetime statistics of the part.

The third strategy is called *predictive maintenance*: instead of estimating the lifetime of a part from statistics, its status is measured. The effects of the deterioration of a part can be compensated for by changing the control parameters of other parts.

To pinpoint the cause of a problem, the first thing which is done is collect information that is available (customer description of the complaint, engine error messages, saved incorrect prints). This information is then stored and / or sent. The information is scanned for anomalies in the analysis phase. This should lead to one or more possible failure causes. To be more certain and to eliminate (most of the) possibilities more information might be collected and analyzed. The last phase is fixing the failure by replacing parts, changing settings, environment or consumables. At the end of the fixing phase, the cycle of collecting info and analysis is often performed again to check whether the fixing was successful.

### Error detection handling and recovery: MRE

When a failure occurs that leads to the machine being internally in an undefined state, the machine is stopped and the Machine Recoverable Error mechanism is triggered: The engine will request the user to reboot the engine. When the failure was temporary (e.g. glitch on IO), this is the fix to get into a defined state again.

### Error detection handling and recovery: ORE

The engine checks during run on regular basis a number of sheet position related items by comparing measured values against the expected values/thresholds. If the measured value compared to the expected value exceeds a certain value then the machine software reports an error. The machine tries to fix the machine by flushing all paper out of the engine to prevent user interaction. After flushing, the machine stops and goes to standby. In case not all paper is flushed out the user fixes the error situation by removing sheets manually (Operator Recoverable Error).

### Runtime contradiction

When an error occurs which can be expected during normal operation (like for instance: out of paper), the run time contradiction mechanism (RTC) is used. When a RTC occurs the printing will stop and the UI shows a message indicating what the reason for the stop is. After the user has solved the error, printing will continue.

### Error detection: Warning

The warning mechanism is triggered when the machine detects an anomaly that does not lead to damage to the customer, damage to the engine or output that is outside specification. The machine will log data of the occurrence in a service log as a warning and continue normal operation. The user does not experience any problems.

Some warnings are an indication for service that a part is deteriorating and that a MRE is probably upcoming and replacement of the part is needed or adjustments should be carried out.

### Error detection handling and recovery: PE

When a failure occurs which cannot be fixed by resetting the engine, a Permanent Error (PE) is reported. This will stop the engine and shut it off. After this the operator cannot switch it back on again. A service engineer has to fix the problem and clear the PE flag before the engine can be switched on.

### Counters (preventive maintenance)

Action counters in the engine can be divided in two types: usage counters and maintenance counters. Usage counters are machine life time counters that cannot be reset. Maintenance counters are resettable and count the lifetime of a replaceable part.

### Logging

Logging is the sequential recording of sensor readings, output levels sent to actuators and observed software states and software communication. The logging can thus provide much information on the machine’s condition over time and is therefore very useful in analyzing a failure. Especially during the design phase, but also thereafter, many involved actors make use of logging.

Logging comes in many variants: ad hoc or long term stored, extracted via a serial interface or via Ethernet or USB. It can be stored on an internal or external file system, either persistent or non-persistent.

Logging is enhanced in a number of ways:

* The moment of logging. In older engines logging of the history was not possible, only logging of the engine with the service engineer present could be made.
* The period over which the logging is made. In the past the logging period was short (shorter than 1 second).
* The number of lines of logging per second.
* Adding more hardware information to the logging. The logging is becoming multidisciplinary. Where in the past logging was used mostly to follow the software states and variables, currently the trend is to log many hardware related parameters.

Due to tolerances, the hardware in the field can differ from the hardware tested during development. It is therefore essential to be able to log the hardware parameters in the field to analyse problems or to check theoretical latitudes in practice.

In the past R&D was the solely user of logging. The service organisation now also uses the logging due to the fact that the logging contains more and more hardware related loglines besides the software logs. Service can use filter mechanisms to get the information that is relevant for him. Critical print process related parameters are for example particular useful for analysing print quality problems.

The REFLEXION project will provide mechanisms to use the vast amount of data in functional logging (that describes detailed physical machine states and machine usage) to:

* Improve the machine design by using increased understanding.
* Limit downtime by using faster diagnosis.
* Guard manufacturing quality and adjustments.
* Support business processes using increased knowledge of machine usage.
* Optimize customer processes using knowledge of machine utilization.

### Input, Output, Special, System test

In case an error code does not pinpoint to a unique unit/part, service can collect additional information by performing a test. These tests vary from simple IO tests to special tests.

### Adjustment test, automatic process optimization

A lot of engines contain mechanisms that can be adjusted mechanically, electrically or by software. The purpose of these adjustments is to compensate for tolerances in the used parts (installation issue) and deterioration of parts such that the functionality performs optimal.

### Handicapping

Some machines automatically handicap parts or features if the same error occurs some times in succession. This allows the user to continue making use of (a slightly less convenient) machine. When a unit is handicapped this will affect the behaviour of the engine. This can have impact on several characteristics like productivity, energy usage, lifetime, etc. Handicapping can be done by the customer (disabling an input tray) or automatically by the engine.

* 1. Lessons Learned

Our original aim was to set up a framework to use logged machine data to support diagnosis, during development as well as during operational use. However, once such an infrastructure is available, its benefits are much bigger. For example, data feedback and analysis leads to a closer relation with clients and their needs and makes it possible to better incorporate client needs in products. Deployment of machines becomes predictable and even waste (like paper and ink) can be reduced. Parts can be used during their complete lifespan and do not need to be replaced anymore at regular intervals.

An example of an important application is detection of deviations from the normal behaviour or monitoring of degradation of parts. This can be used for predictive maintenance, where one tries to predict whether an error will occur in the next period (for example a day, two days, a week, or a month). In this way one can notice a problem and plan service visits before the effect really surfaces. Machine learning on the acquired data is a helpful tool, but needs (printer) domain expert guidance to select the relevant features.

Other applications are dashboards with real-time reporting on productivity KPI’s, possibly on multiple systems (in this way service and customers can monitor multi-site fleets) or finding out how a user really uses a machine.

The fact that we have chosen to use structured logging of functional data, which is consistently timestamped, has been proven to be very successful. The data is ordered in a tree according to the machine structure and has a tabular form (not a free textual logging format that is often used in a software context). It is also important that the data format is stable over time (no ad hoc addition or deletion of columns), because the analysis might otherwise be hampered by inconsistencies.

For the analysis we do not want to rely on data scientists only. The idea is that function engineers, who are knowledgeable on the system they are developing, can perform their own analysis. The Jupyter notebook environment combined with Python (and its data analysis and graphics libraries) and a number of custom made user-friendly log data fetching and cleaning utilities proved to be very useful for such people who are domain experts, but not data science experts.

This ODAS system, in which domain experts can develop notebook code for data analysis, is easy to learn, and helps them to solve a typical data science question in a few hours. Notebooks are shared between users and alerting functionality gives triggers based on notebook output. The technology is open source: HDF5 contains tables in a hierarchical tree, Python is the main notebook language, pre-processing uses Pandas, interactive visualization is done with Bokeh, and data processing with a.o. Scipy and Scikit-learn.

A special internal ODAS course has been developed. The duration of an instance of this course is five afternoons. There is a special homologation for domain experts without programming background (although they often have for example some Matlab knowledge). During 2018 more than six course instances are given.

Once many people are developing and using notebooks, they should be able to retrieve notebook code that has already been developed rather than being forced to “reinvent the wheel” over and over again. For this we are performing research on assisted notebook composition. Here, the starting point is a non-formal high level question from a domain expert like ““do sheet rejects relate to media type for all printers in the USA?”. From this question, one derives more formal detailed questions like “select all USA printers”. The existing notebook cells are searched for matching content and the search results (code snippets) are used to compose a (partial) new notebook that will be manually refined. Simple search approaches are straight forward text based, or use log file and log folder structure. Document-based semantic approaches (word co-occurrence, Hearst patterns, depparse) can help to find words related to the search terms.

1. Philips IGT
   1. State of the Art

### From X-ray system to Intervention solution

The portfolio of Philips IGT systems changed for an x-ray product to end-user solution in the last years. The focus of Philips IGT systems changed from fixed x-ray systems imaging system to a cardio vascular intervention solution. At the intervention site a major step in this strategy was the acquisition of Volcano. This provides a suite of peripheral therapeutic devices which enables physiology and intravascular imaging for coronary and peripheral applications. At the imaging site the X-ray system has become integrated node in the diagnosis workflow. The system enables to connect and integrate several third party systems for the diagnosis and post-processing of image data. Furthermore it is an integrated part in the hospitals IT system in the handling of patient data.

For maintenance and to support the previous described strategy a remote monitoring system is already in place to serve the customers of Philips with their medical systems. Our Customer Service (CS) department is responsible for this service to collect, view and store system logging of MR, CT and X-ray systems. They also have the ability to retrieve a logging in demand instead of daily or to share the screens of their device with the CS helpdesk.

The scope of the REFLEXION project is therefore not to optimize the event log acquisition or content. In this project Philips IGT systems is aiming to further integrate the information of the system event log into the verification process with in Research & Development.

### User profiling in system test

Before a new X-ray product can be released by R&D its specifications needs to be verified and validated. Ideally this verification and validation testing is performed in the context of our clinical user. During system validation, for example, Philips IGT systems features special validation bays. These test environments contain all equipment of a regular cathlab and also look like a cathlab to simulate clinical procedures.

To ensure that we deliver high quality reliable systems to our customers we verify the reliability of our systems during system verification. The goal of reliability testing is to drive a system with realistic clinical scenarios in conditions which were not expected on forehand, taking care that the system is not load and stress tested. To evaluate the reliability of a system the response of the system has to be monitored during the test time. If the response is not as expected, than the system made a failure which can probably impact the user or the patient. Reliability figures are usually expressed as Mean Time between Failures (MTBF) ranging from a few hundreds to thousands of hours. In practice this means that a few thousands hours of testing should be performed to verify its reliability. Therefore reliability verification is an ideal candidate for automatic testing.

### State-of-the-art of user profiling

To enable verification with clinical scenarios in-house software was developed to use event log data from the installed base. This resulted in a software tool ProMise that is able to perform process and data-mining on event log data. Based on selection criteria e.g. system type, location, time period, etc. general statistics are presented based on the event log data of the selected systems. ProMise can then provide specific information on, for example, the number and type of procedures, errors and system utilization. From this data process mining can be performed to visualize the utilization of the selected systems.

The process and data mining of event log data with ProMise provides a valuable step to support the use of clinical user profiles for system verification. However, ProMise is lacking the possibility to provide a typical user profile which represents a generic user profile. Within this project we aim to develop methods to generate typical user profiles which will be integrated in the verification process.

* 1. Lessons Learned

The focus of Philips IGT during the REFLEXION project has been to improve the efficiency and effectiveness of the System Integration and Verification through data analysis, model-based testing and machine learning.

Data processing and providing insights

Increasing the data volume inherently results in longer computation times. On the one hand, scaling up computing and memory resources can mitigate this. We are moving our ETL scripts from individual machines towards a Hadoop cluster, thereby greatly increasing computing power. On the other hand, we have increased the performance of our dashboard explicitly by creating a highly compressed buffer of the data on our dashboard server. Data is compressed by a factor of ~100 – 1000x by reducing redundant data through grouping of the data in a smart way. Also, a star schema database structure, a structure in which the centre of the star contains the linking attribute for all other applicable tables, has been applied leading to simpler queries, faster aggregations and better query performance. Last, we applied a columnar database structure to reduce the amount of data that needs to be read by efficient compression and by reading only the necessary data from disk.

In order to provide users with insights in the logging, it is important that the data is abstracted to some extent (Figure 8). Event logging is often full with low-level events and sensor data and while it is possible to extract all this information, adding a filter and an abstraction layer results in the processed data being much more user-friendly. This abstraction layer is designed by domain experts with insight in clinical practice, who are able to provide context to the low-level events. Defining the translation table needed to create the abstraction layer is an iterative process with close collaboration between domain experts and the ETL developers. By making this abstraction layer, we have created a dashboard that allows users an overview of system functionality, but also provide means to deep-dive into the relevant metadata of these actions.

Especially during development of the Extract, Transform, Load (ETL) scripts, many iterations are required to optimise and debug scripting. While the domain experts could define the mapping of many actions, the developers found many exceptional event combinations after initial log processing, making multiple iterations required. As these scripts often take a long time to complete processing (because of the need to go through every single line logged again) we have chosen to develop on only a limited, but representative, selection of worldwide systems. This allowed for a much shorter processing time, resulting in faster feedback thus decreasing the development cycle time.

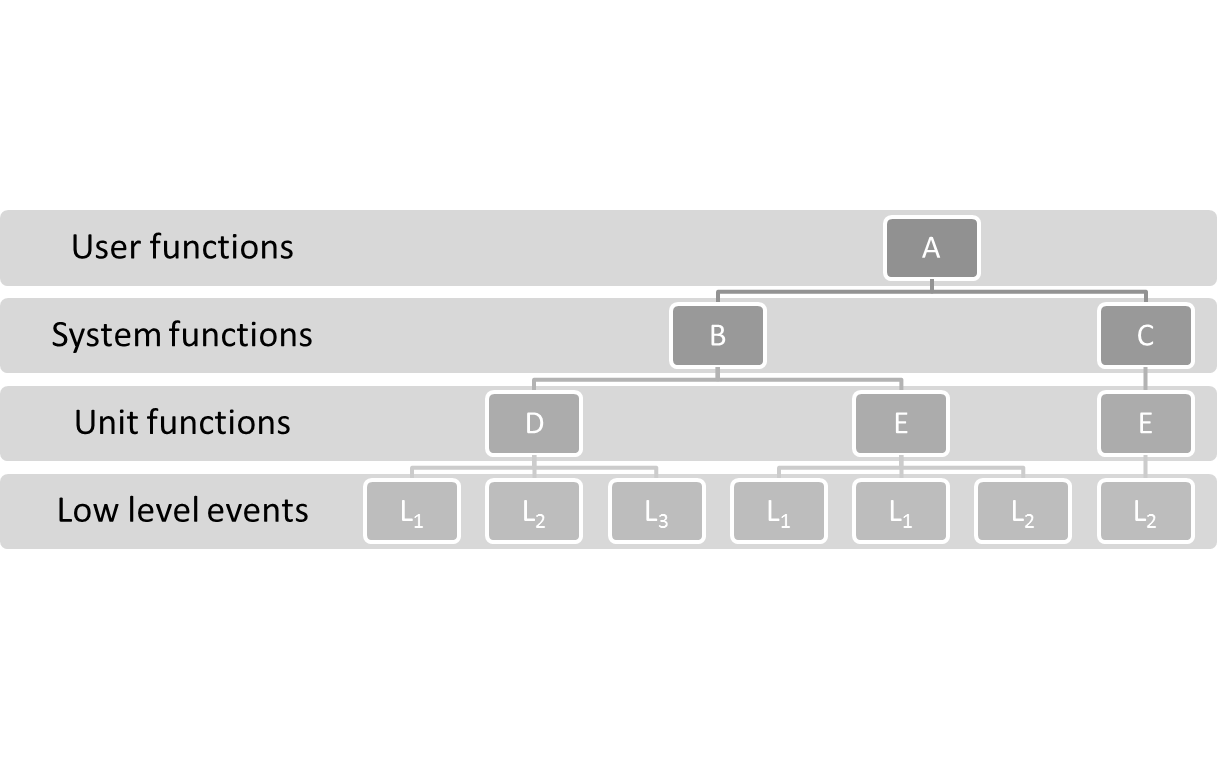


Figure 8: Log granularity and abstractions levels. The profiling is usually required from the point of view of the user thus available logged data contains information about low-level events.

One of the most important aspects of making sure the log processing scripts remain up and running in the years to come, is by placing constraints on the format and content of the event log. The only way to achieve this is by creating the notion that the event log is more than a debugging tool, but rather an invaluable source of information on how our customers are using our systems. By placing constraints on what and how developers can write to the event log, we can make sure that the ETL-script remains compatible with the event log for future system releases to prevent doing twice the work. However, it is inevitable new system functionality will cause changes to the event log content, making prolonged support for the ETL-script necessary.

While the initial approach in REFLEXION was to use the processed field usage data for the creation of usage models for model based testing, we have seen a large interest for the data throughout Philips IGT. Different teams have shown interest in the created data analysis dashboard, and first initiatives to use this data have emerged. For example, the data is now being used for training purposes of new employees within the System Verification department, as it shows which actions are performed during different clinical applications. Complemented with the clinical context provided by application specialists, this gives new employees a great basis for understanding the importance of certain system functionality. We are confident this training material will prove its value throughout R&D. Also, employees without SQL knowledge now have a fast and easy way of visualising what happens with our systems in the field.

Another example of the benefits of having this data at our fingertips, is that the System Design team has been able to base decisions regarding our UPS unit (Uninterruptable Power Supply) based on our data. By getting insights in how often high-power actions are performed, we can fine-tune the required size of our UPS units.

Hierarchical Usage Profiling and Model-Based Testing

During the development of our model-based testing (MBT) pipeline, we have shown that we are able to set up the infrastructure needed in order to get data from our data lake, through our dashboard, to Axini’s Test Manager where test scenarios are generated which could then be automatically executed on our systems. Test case generation purely based on frequencies of actions has proven not to result into realistic test scenarios, demonstrating the need for additional intelligence, provided by either the domain expert or through data analytics and/or machine learning. Machine learning is the method of choice here as domain experts are often limited in the number of dependencies between actions they can think of, and the dependencies may change depending on the clinical application we are trying to simulate.

By nature, the models generated by process mining are susceptible to ending up in indefinite loops. However, Axini’s Test Manager can be used as a controller of the paths through these models. The combination of the learned models in combination with Axini’s Test Manager has been identified as a promising opportunity of using field usage data as a basis for MBT. More investigation is needed to determine the feasibility of this approach.

Also, while it was feasible to generate test scenarios based on the user action statistics, the hardest part of this task is to incorporate the metadata of these actions. With many different mechanical movements being possible, it becomes crucial to define the boundaries in which the different parts of our system can move freely in order to prevent collisions. The classical engineering approach would define a large ruleset in which all movement ranges and dependencies of movements are described. An interesting topic for follow-up would be to design a method, e.g. by reinforcement learning, which knows or learns which movements are possible and which not, and if the latter is the case how to resolve this.

Usage Profiling and Machine learning

A quick online search on machine learning results in a plethora of different publications on available approaches, techniques and algorithms. However, even the newest and fanciest algorithms are not plug and play and require careful consideration of the task at hand. After that, log data needs to be pre-processed into a suitable format, often a tedious process in which a lot of domain expert knowledge required. Later, many iterations of validating the algorithms outcome and fine-tuning parameters are needed to hopefully arrive at a satisfying result.

There are many different measures available to quantify the accuracy of machine learning algorithms, such as mean squared error, fitness functions or the deviation from uniform distribution as used in our case. One thing these metrics have in common in that they show the mathematical optimisation score of the applied algorithms, while they are not able to judge the semantics of the generated models. For now, the only way to assess this is by asking experts to judge the models, which is a challenge on its own because of both human and technical factors:

* experts can be knowledgeable on a very specific aspect of machine usage
* experts are biased towards how they think the machines should be operated;
* other experts can have different opinions on the same topics;
* experts, when observing usage model graphs composed of many nodes and edges, might encounter difficulties in analysing them;
* experts can only perform indirect validation for non-trivial models.

We have overcome, or at least reduced, these issues by means of a crowdsourced variant of the famous Turing test for artificial intelligence. In a nutshell, we created a dataset composed of *real* clinical procedures – i.e. procedures retrieved straight from logs – and *synthetic* procedures – i.e. the traces generated by the usage models. We then asked a simple yes/no question to the experts: “*does this sequence of actions occur in clinical practice during clinical practice?*”. Because we showed the experts not the actual usage models but rather traces through these models, we allowed them to judge N-grams with N>2 as well as the word vectors which were a result of the doc2vec algorithm.

It remains an interesting topic how to accurately represent N-grams with N>2 for expert validation, without having to create excerpts of it. Another important questions that we can ask after performing the Turing test is how can we rely on the experts opinions, given that they are not able to achieve a perfect score in a test where only real clinical procedures were shown.

One of the goals for REFLEXION was to improve the verification process of System Reliability. Initially it was thought that MBT, Process Mining or Machine Learning would result to methods for generating artificial test scripts. However, we have seen that probabilistic procedure modeling is much more fit for event log classification rather than for synthetic test case generation. With this approach, we can identify which patients were performed with a “normal”, “odd” or “exceptional” workflow. Depending on the needs, logging of a specific patient can be selected and converted into a test script. By creating an optional expert feedback loop, the models can be trained and fine-tuned for optimal log classification.

Connecting the dots

A general finding is that besides the data scientist and engineer, the domain expert is a key player to make these algorithms and methods to a success. From start to finish domain experts are needed in defining the mapping between low-level events and system functions, defining the boundaries of procedures and chunks actions, semantically evaluating probabilistic modeling outcomes, etc. It is therefore an absolute must that experienced people, with in-depth system knowledge and a well-developed network, are selected for participating in similar project.

As the event log file is a crucial component within our solution, it is of utmost importance that the quality of the logging is guaranteed. We need to ensure that it is part of any new system project. Missing adequate logging on a single system release, results in a future event log gap of approximately a year.

In order for the project to deliver a product that will be adopted by the organization, it is of utmost importance that stakeholders are involved at an early stage during design. This ensures that a *pull* is created from within the organization, and that the designed product is in line with the expectations from the stakeholders. Another aspect to keep aligned with stakeholders is that starting small and developing prototypes optimizes the chance on success. Expectation management is also highly relevant between the different partners, and partners need to be open on how much and where time is spent.

Also, we have seen the benefits of creating a workplace where all involved partners in the project can sit together. This greatly improves communication between team members and speeds up learning to speak in the same language, which is highly relevant as we have observed that similar terms, e.g. the terms “model” and “functional testing”, are interpreted differently by data scientists, MBT experts and domain experts. Another aspect which was of great influence were the workshops organized within the REFLEXION project, where experts with different backgrounds had the chance to discuss, align and transfer knowledge.

1. Philips MRI
   1. State of the Art

In this section the way of working and features are discussed, assisting in improving insights and requirements of Philips MRI scanner at the start of the REFLEXION project. Philips MRI scanner are used in a hospital setting to create images for diagnostic purposes. The demand on Uptime is high. Uptime is defined as the ratio of time the system is up and running compared to the time the system is down due to failure. Reliability and Serviceability are key elements to provide the required uptime which is close to 100%.

To guarantee and improve Reliability insights in the performance of the systems installed in the markets is required. Insights are collected by reaching out to the customers asking for feedback. Insights are also retrieved from machine data.

### Way of working

Insights from the field are retrieved by customer complaints and direct interaction with customers. Customer complaints are registered entries submitted by the local service engineer in most of the cases. The local service engineer has a close contacts with the customer. The service engineer meets the customer between 2 and 12 times per year, depending on the demand of the customer, the state of the machine and the utilization of the machine by the customer.

Customer complaints are accompanied by log files and a description of the problem. The log files are analysed by the complaint handling team and when corrections or bug fixes are required the issue is assigned to R&D to implement and program the solution. The R&D engineer will analyse the problem in a more depth. For cases which are obviously wrong, the log file accompanied with the complaint is sufficient to find the root-cause. In case of intermittent problems or issues in system budgeting, more facts and statistics from the field are required. The R&D engineer will collect log files from a shared drive which contains the log files of most of the connected systems and writes analysis scripts to find similar cases over a small selection of the available log files. Tuning the scripts and re-processing the logs has typically and iteration time of 2-5 days. With the analysis the R&D engineer tries to get insights in the distribution of system performance to check the facts from the field with the theoretical models made in house.

Insights are also retrieved by direct interactions with the customer. Especially the applications R&D engineer gets in touch with selected customers to discuss if the protocols delivered with the system match the customer’s need. The insight is used to update the protocols and drive future requirements in case further optimizing the protocols is not possible without a system change. At this moment the application engineer has no insight in use of the delivered protocols over the complete installed base. Improvements on the protocols are triggered by the discussions with customers and based on feedback received via the complaint handling unit.

### Features

Features used today to perform analytics as described in the previous section are described in this section.

* Unstructured machine data: The MRI machine today generates 100 – 200Mb unstructured time series data per day per device. The information is stored in flat text files. Sensor data is stored in separate files per sensor. In one central log file all events of all subsystems are stored. Each log event in the device is added to the file as new line. This log file is used to place all parallel activities on the multiple cores and hardware components in relation with each other. The log files are initially created for usage by development for debugging purpose. Today the log files are used by service to diagnose the system.   
  The information in the files differs per release as there is no formal requirement on the format nor on the information logged guaranteeing consistency over releases. The machine data does contain the most relevant user actions like starting scans. Not all user actions (like modifying protocols) are logged.
* Collect machine data: Today the MRI machine data is stored on the device for 1 to 6 months maximum. For a subset (70%) of the installed base the data is retrieved daily to a central repository. The success rate of downloading is around 70%. This means that for connected systems around 70% of logs are centrally available. The data at the repository is kept for 1 year. Data privacy has taken into account when connecting and collecting machine data. Patient data like images and Patient names are not transferred.
* Log browser: A log browser application is available supporting quick search through the log file using regular expressions. This tool is used by development and service to find patterns in the log files.
* A website is available where development and service engineers can subscribe for log events. Each log file transferred from any connected MR system to the central repository is parsed for the subscribed log events. Notifications to the subscribers are given by mail in case matches are found.
* Dedicated parsing scripts: Various parsing scripts are available, developed by R&D engineers. The scripts typically reads the logs from the central repository and put the output in csv or xml format. The output is further analysed using Excel or Matlab.

### Delta with vision

The major delta between the vision and the situation today is the speed of the feedback loop for data insights in product reliability improvements. Where today multiple iterations are required each taking days for collecting & visualizing data for a selected set of systems, in the vision this should be reduced to hours or a fraction of it. It is the challenge in the REFLEXION project to address the key elements in the chain from machine data creation till data visualization.

* 1. Lessons Learned

During the REFLEXION timeframe a framework has been implemented and released which collects & visualizes machine data within a latency of 2 days for all connected MR systems. The unstructured raw machine data is converted in structured data, translated in information by analytical engines and showed in dashboards. Experienced developers are able to query the stored structured data for their own analysis purposes.

The risk of the framework lies in the availability and the reliability of the data. The availability mainly depends on connectivity of the MRI systems and depends on IT framework techniques. The reliability depends on the stability of the machine data (the logfiles) over releases, the quality of the code writing the logs and scripts extracting the information.

One of the key elements which enabled the successful implementation was to make the R&D development department owner of the ETL (Extract Transform Load) scripts translating raw unstructured machine data into structured data which can be loaded into a database. By making R&D, the creators of the machine data – the logfiles, owner of the ETL, the R&D felt the need for a sustainable solution. The logfiles became a formal interface of the product and were optimized for ETL parsing.

Visualization and accessibility of the structured data enables fast and fact based decisions. The data is now shared in project and management reviews as facts, shorting the lead time to solutions. For example on the area of reliability, regression tests are executed with operational profiles based on the machine data retrieved from the field. The test systems are monitored on the trend of errors and unexpected behaviour. Those reliability facts are plotted during the project meetings and actions are taken based on the facts. The scripts extracting this information are fine-tuned during the verification period and re-used to pro-active monitor the new systems in the field during the monitoring period shortly after release. Dashboards are daily updated and allow a quick drill-down to the root of the issue. This shortens the plan, do, check, act cycle significantly. The structured data is also used as input for Weibull plots and provide insights in the reliability trends and failure modes of hardware components.

Visualization commonly starts by experts making plots with excel, mathlab or notebooks. Once interesting items are found, dedicated visuals are created using off the shelve tools. During the REFLEXION project we experienced the power of those off the shelve visualization tools. The tools enable to bring the insights of the expert to a bigger audience. The tools allow other users to drill in the data with a few mouse clicks. Once the interested audience is growing the visuals are embedded in an online dashboard, making the visual available for everyone in R&D and are auto updated every day.

Not only the R&D department benefited from the framework and availability of structured machine data. The data is used by Marketing to check whether new features are adopted and utilized by the customers as planned. The data is also used by the service department for sending out proactive service alerts. Even commercial Customer facing utilization dashboards are developed and sold as service to the customer based on the same data. As all disciplines acknowledged the added value, additional non-machine-data ETL’s, parsing process data, were added to the same framework. This opened a new landscape of analytical opportunities. Parts can be tracked and repairschop data can be connected to correlate machine behaviour to failure root-causes from the repairshop. This enabled faster design improvement cycles and faster serviceability improvements.

During the analysis of data it became clear that Data Scientists as well as domain experts are needed to translate the huge amount of available data in actionable items. Domain experts are needed to filter data at the input and classify the relevance of the outcome of the analysis and steer the next iteration. Data Scientists are needed to use advanced state of the art techniques to mine, process and visualize data. Only when the two disciplines work closely together the chance of actionable results is significant. The challenge is to create a concrete questions for the Data Scientist. Questions like “provide me an aggregated view of all MR scans performed” is an example of a non-concrete question leading to non-actionable results. When the skills of the Data Scientist and Domain Specialist overlap, the chances of actionable results increase. At MR R&D a MR-Eye Ambassadors group has been formed to exchange Data Science, and Domain Knowledge information.

1. Siemens ISW
   1. State of the Art

Various mechanical and electrical faults develop during the course of machine operation in the mechanical industries whether due to natural wear and ageing, defective parts, or extenuating operating circumstances. These faults decrease the overall machine performance and, if left unaddressed, they can lead to a complete system failure and shutdown of the machine. It is therefore important that the “health state” of the machine is monitored during operation so that the developing faults can be detected and diagnosed before they lead to significant damage, costly repairs and extended unplanned machine downtime. So-called “Condition Monitoring” (CM) techniques [4, 5] indeed achieve this by monitoring sensor signals (i.e. vibration, temperature, etc.) which are acquired during the machine operation. A further step is to “predict” the remaining useful lifetime based on the combination of operational measurements and models, such that predictive maintenance actions could be undertaken.

Predicting the safe functioning life of mechanical structures and components is a central topic in the field of mechanical engineering. Yielding, fatigue, and fracture are all possible failure modes that can be activated by time-varying loading, and numerous models and constitutive equations have been proposed to describe failure under these conditions. These criteria have been shown to provide accurate predictions of component life if the material properties have been precisely characterised. They are particularly effective in the case of structures with simple, smoothly varying stress states such as beams, plates, or pressure vessels. Predicting the life and symptoms of failure for components with complex geometries or multi-body interactions (e.g. gears, bushings or bearings) presents significant challenges. Active academic research on improving estimates of life and modelling failure modes of these components continues to this day. The challenge of lifetime prediction is further complicated by the time-varying operating regimes and dynamic responses with frequency spectrums spanning orders of magnitude. Models proposed in the literature have widely varying complexity in terms of number of parameters and degrees of freedom; however, the necessity of additional complexity when the model is to be used for automatic failure signature recognition using machine learning algorithms has never been investigated, and it may be that a highly simplified model is sufficient, or the most complex models are insufficient. This questions will be directly addressed in this project as models are used to train machine learning algorithms.

Machine learning (ML) algorithms are classified as such because they are able to adapt to and to make predictions from data without explicit programming. These algorithms are currently the most successful in complex tasks in which traditional techniques have struggled, such as image classification [6], speech recognition [7], and artificial intelligence [8]. Within the domain of mechanical engineering, machine learning has seen the greatest usage as a data mining technique to predict system performance and failure using in-service data [9, 10]. Many applications do not benefit from a large database of historical data, and therefore the feasibility of training ML algorithms with model-generated data to predict physical failure is of important industrial consequence. The state of the art of machine learning applications with regards to possible failures to be considered in this project is detailed here.

### Condition monitoring in mechanical industries

Condition Monitoring (CM) techniques exploit the (large amount of) sensor data which is acquired during the machine operation. This (for example) allows for fault detection of the monitored machines. In the current state-of-the-art, there are however following challenges which require further innovations:

* In many industrial cases a so-called “Machine Fleet” is deployed, i.e. a fleet consisting of multiple similar or even identical machines. Examples are identical drivetrains in a manufacturing plant [11], a wind farm consisting of multiple wind turbines [12], and a fleet of aircraft or vehicles [13]. Current CM solutions do not fully exploit the fact that there is a fleet of similar machines, i.e. the methods only make “vertical” comparisons (over time) on one machine [11]. A key goal in this project is to investigate in what sense the machine fleet concept also allows for making “horizontal” comparisons (“machine-to-machine”). The added value could for example be that anomalous behaviour (of one or more machines in the fleet) can be more easily and more quickly detected, compared to the single machine case.
* An important challenge is that, even though the machines may be completely identical, they are typically deployed in very different working conditions, e.g. because they are manufacturing different products [11]. As a consequence, multiple “operating regimes” are encountered, i.e. the different machine drivetrains may be operated under different possible combinations of rotational speed, load, temperature and other parameters [12]. An automatic determination and clustering of the operating regimes is then an essential prerequisite for successful fault detection, because feature differences due to faulty behaviour have to be discriminated from feature differences due to differing operating regimes. Moreover, the operating regime determination is also an important goal by itself, as it provides important insight on the actual machine fleet usage.
* In addition to the fact that the machines drivetrains are operated in multiple operating regimes (cf. supra), certain operating parameters also vary considerably during usage [12], so that for example the measured vibration signals should be viewed as realizations of non-stationary stochastic processes. The traditional, state-of-the-art CM solutions [4, 5] are not adequate in such applications, as they are typically based on the assumption of stationary operation, such as a constant load or a constant rotational speed [12].
* Another limitation in current CM solutions, which often prevents successful deployment in practice, is that they typically lack adaptability and self-learning capabilities [11], so that experts are required to implement (and re-train) the system for a particular application. For example, a “rule-based” approach may be deployed whereby the health condition depends on whether certain parameters exceed (human-defined) [13]. The challenge is to move towards true “self-learning” methodologies, which are generically applicable in a variety of applications, so that such manual tuning is no longer required.
* A wide range of motor-sizes is typically encountered in industrial drivetrains [14]. Similar to the previous point, this implies that any researched methods should be self-learning and generically applicable so that they can automatically scale up with motor power rating.
* A final challenge is that, typically, only historical data on “normal use” may be available, i.e. it is difficult to access historical data acquired in unhealthy or faulty conditions [15, 16]. This implies that unsupervised learning methods, whereby it is not required to have representative training data for every possible condition (e.g. for both healthy as unhealthy conditions), are most appropriate. As these methods do not rely on training, they may also be more adaptable to different working conditions (cf. supra), which is an additional benefit.

### Modelling Component Failures

The failures modes summarized in this section are limited mechanical and electrical components common to rotating machinery and automotive applications. Certain failures are particularly challenging to predict using remote sensing techniques, such as shaft failure. Yielding is unpredictable as it depends only on the instantaneous loading and material properties. History dependent processes such as fatigue failure can be predicted using traditional stress or strain based techniques such as Basquin's equation and the Manson-Coffin relation for low cycle fatigue, and fracture can be predicted using the Paris crack growth law [17] with an assumed initial crack size; however, sensing the deterioration associated to these failure processes requires detecting small changes in shaft stiffness that occur because of cross-section reduction by crack propagation. These challenges, combined with the relative ease shaft design and manufacturing, precludes their analysis in this study. Components with intrinsically time-varying behaviour such as gears and bearings are far more complex to design, manufacture, and manually inspect and therefore could benefit much greater from an advanced failure analysis using machine learning. In this section, the most recently proposed models for these components are reviewed.

#### Gears

Gears have existed for thousands of years, and yet the complexities of gear dynamics continue to drive modern research efforts to better understand even the simplest of gear geometries (spur gears). The dynamics of gears was first considered using one dimensional spring-mass models before moving to a more complex accounting for gear tooth stiffness and nonlinear effects [18]. Modern models consider phenomena such as variable tooth stiffness and Hertzian contact stress at the contact patch between gear pairs. Beyond undamaged gears, numerous models have been proposed for gears containing defects with emphasis on tooth root cracks and gear face damage. The next several paragraphs summarize the state of the art of spur gear models with defects.

Cracks can occur at the root of gear teeth for various reasons including flaws in the material or crack initiation sites created during manufacturing. The loading state at the gear tooth root is characterized by a stress concentration and fatigue loading, which provide favourable conditions for crack growth. Consequently, gear tooth root cracks are common defects for gear teeth as shown in Figure 9. As the crack initiates and propagates across the tooth root, it reduces the cross-sectional area of the tooth connection at the root circle and this causes changes the gear mesh stiffness when that tooth is engaged before the tooth is snapped free from the gear circle. The change in stiffness is propagated into the dynamics of the system, which is how this type of defect can be detected.

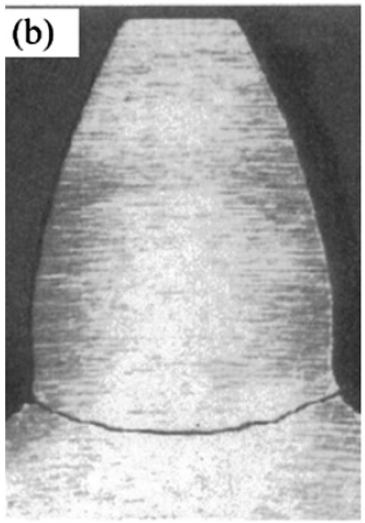


Figure 9. Gear root tooth crack. [19]

Yang and Lin [20] developed an analytical expression for gear mesh stiffness for a gear containing a straight root crack using an energy method, which was expanded by Wu et al. [21]. Chen and Shao [22] extended the model further to include a straight crack at an angle, and Wan et al. [23] included fillet-foundation deflections and flexibility in the gear circle. The models are often compared to experiments within the framework of a larger model including dynamic effects from connected components such as shafts and bearings. As a result, different studies are difficult to compare and the additional degrees of freedom and assumptions may introduce errors into the analysis. Some studies include up to 26 degrees of freedom [24]. What has lacked investigation is the accuracy of gear models as the complexity is increased, and the capabilities of the models with regards to physical failures. Beyond first principles modelling, numerical methods are commonly applied and are well regarded as accurate representations of gear behaviour. Finite element method (FEM) calculations of gear mesh stiffness with a root crack have been performed with elasto-plastic fracture mechanics [25], extended contact models [26] and other crack types [27]. Such calculations can provide valuable inputs to reduced order such as a variable gear mesh stiffness’s or characterisations of stresses in the gear tooth under contact. The drawback of numerical calculations is the extension of the results beyond the simulation configuration. A calculation provides valuable insight for a single gear geometry and operating condition, but lacks utility if the result cannot be modified to apply for other gear sizes, loads, or operating speeds. In summary, the symptoms of gear tooth root cracks can be adequately modelled either from first principles or numerical analysis to perform a comparison to in-service behaviour.

Gear tooth surfaces are subject to harsh loading conditions with contact stresses on the order of gigapascals and sliding contact between metal surfaces. Material or manufacturing defects on the gear face act as crack initiation sites and cracks can propagate and return towards the surface under the harsh loading. As a result, material pitting occurs where fragments of metal are removed from the gear face and contaminate the lubricant. The discontinuity on the gear face causes a stress concentration and favours pitting leading to increased lubricant contamination and gear damage. The complexity of this phenomenon renders it beyond closed form analysis at the present time, and few attempts have been made to model the formation process. The spalling process itself has been numerically modelled using advanced FEM methods [28, 29], but the impact on gear tooth properties are less well characterized. Rincon et al. [30] modelled gear pitting as an elliptical inclusion and used an FEM calculation to determine the change in gear mesh stiffness. However, the authors did not investigate the dynamics resulting from this change in stiffness, and whether the change is even detectable. As shown in Figure 10, the change in mesh stiffness for a 2mm wide, 0.5mm deep pit in a gear face with a width of 6.35mm is on the order of a few percent. The additional frequency components that one could expect to observe may be lost in the noise background. Chaari et al. [31] compared qualitative, quantitative, and FEM characterizations of the effect of gear pitting on stiffness and did not find a clear superior method. Gear pitting therefore presents a challenge to model, and can be reasonably expected to be a challenge to measure.

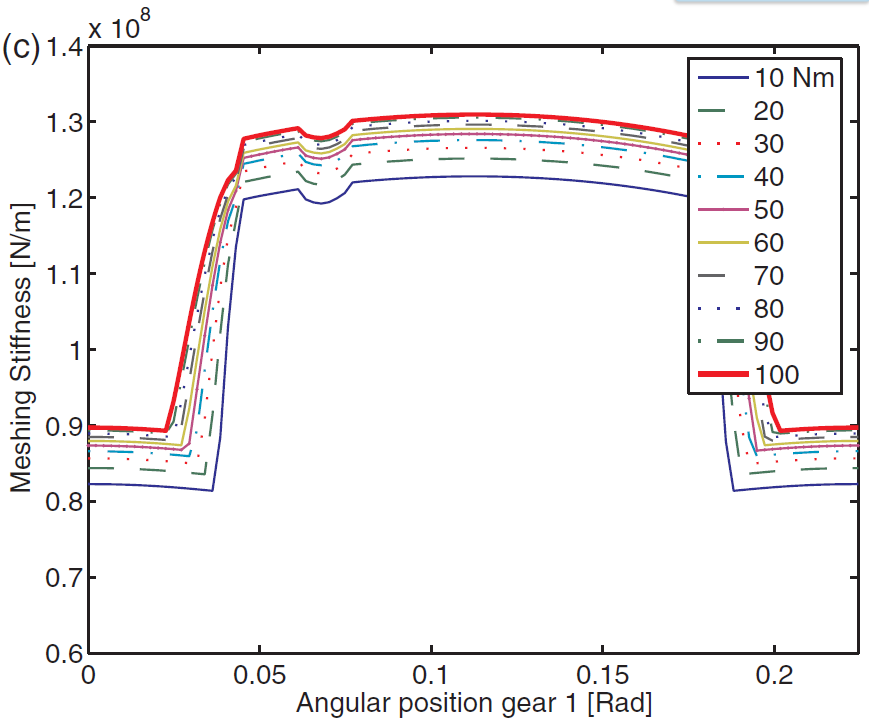


Figure 10. Gear mesh stiffness as a function of applied torque with a pit at 0.065 rad. [30]

Planetary gearboxes appear in numerous applications such as automotive, helicopter, and wind turbines and therefore represent a critical component for analysis. However, the dynamics and vibrations of multiple coupled gear pairs results in a vibration response that has seen few modelling efforts with limited comparisons to experiments [32, 33, 34]. The fidelity of the models is not expected to be sufficient to train an ML algorithm to classify physical failures, and therefore planetary gearboxes are not considered.

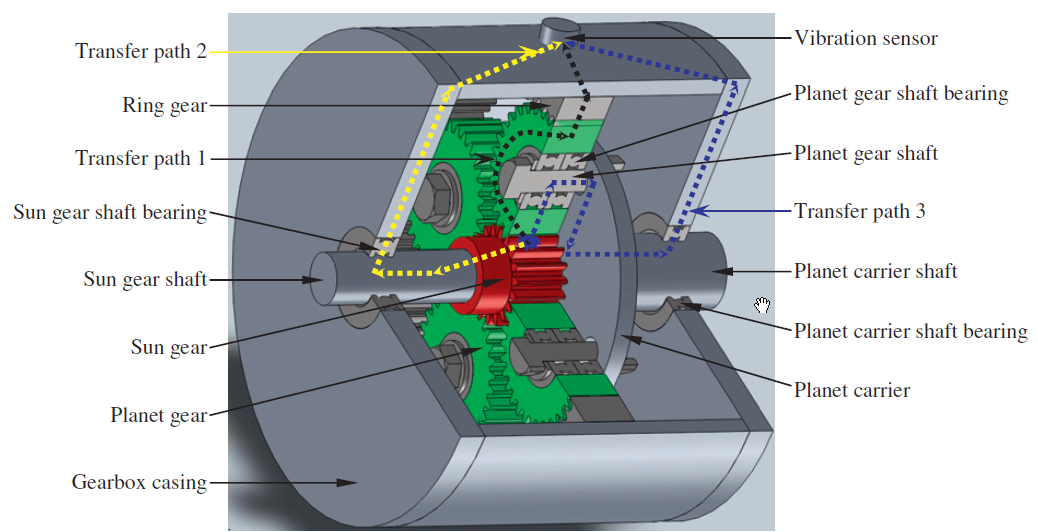


Figure 11. Planetary gearbox vibration analysis is complicated by multiple identical sources with time-varying vibration transfer paths. [33]

Other gear geometries such as helical, bevel, worm, hyperboloid are also not the within the scope of this investigation, as these geometries represent higher order extensions from the basic spur gear configuration. Spur gears are the focus here, as they are the one of the most common gear types and represent a first step towards more complex gear geometries.

#### Roller Bearings

Roller bearings are ubiquitous mechanical components appearing in every rotating mechanism. At a general level, a roller bearing is composed from an outer race, an inner race, roller elements and a carrier. The inner and outer races form the structure of the bearing, the roller elements enable relative motion between the races, and the carrier maintains an equal distribution of roller elements around the bearing. As defects are initiated on the rollers, the inner race and the outer race, the bearing becomes a source of mechanical vibrations and power transmission efficiency is decreased before total failure occurs. Detecting small defects can pre-empt these losses and is therefore an important consideration within the framework of this project. The geometry of roller bearings is complex, as in the case of gears: rollers make rolling Hertzian contact with two surfaces, and a lubricating material can form a film between the rollers and races. Consequently, several different regimes of gear operation can be observed as the lubrication fluid dynamics change [35]. The defects, however, have been modelled with a relatively simpler methodologies than gears.

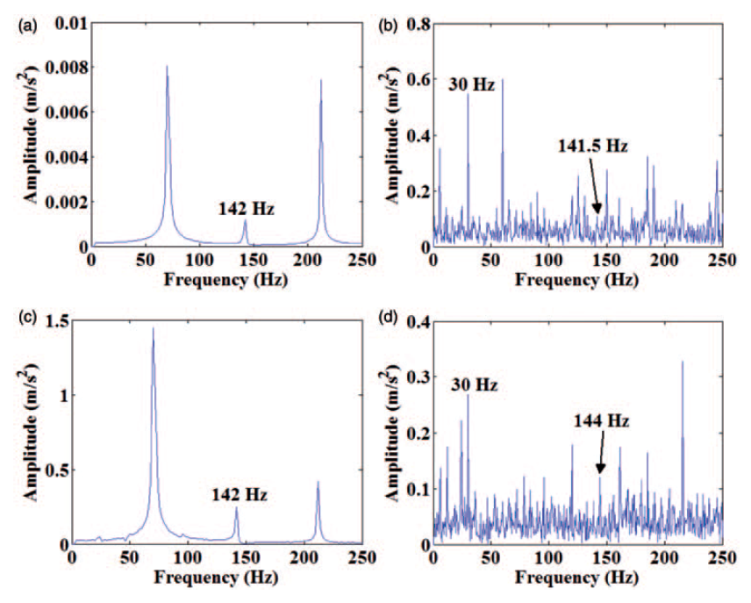


Figure 12. Frequency spectrum for: simulated healthy (a), damaged (c) and experimental health (b) and damaged (d) bearing. It is unclear if the simulation is sufficiently representative to train an ML algorithm. [36]

The first roller bearing defect model was proposed by McFadden and Smith [37]. The response was modelled as a pulse train modulated by the force distribution of a radial load. This simple model has been advanced in more recent years. Brie [38] observed a quasi-periodic signal that cannot be explained by the pulse train approximation, and augmented the basic model to account for a time-varying transfer path for an inner rotating race defect and described the bearing using a single degree-of-freedom mass-spring-damper. Sassi et al. [39] developed a 3DOF model including the bearing loading distribution, bearing structure elasticity, the elasto-hydro-dynamic oil film characteristics, and transfer path effects. However, this model requires calibration to experiments. Finite element calculations have equally been applied to predict the dynamics and vibrational spectrum of a healthy and damaged defect such as in the case of Shao et al. [36]. This model included numerous higher order behaviours such as contact stiffness between the roller and defect, effects of defect width and depth; however, the final outcome of the model prediction was an imperceptible difference between the healthy and damaged roller bearings. Detecting bearing failure from model estimations may be limited to relatively severe damage such that the dynamic response exhibits symptoms above the threshold for detection.

Tadina and Boltezar [40] simulated a full roller bearing using a simple FEM model of beam elements and spring-mass models for roller elements and showed that clear peaks occur where predicted based on the relative motion of the races and rollers. This suggests that an efficient approach may simply be to detect the power of the spectrum around peaks at the expected frequencies and use this measure to predict failure.

#### Induction Motors

Induction motors are simple, efficient, and cheap to manufacture, which explains their popularity in industrial applications. Manufacturing faults and severe loading can cause damage to the stator or rotor and failures inevitably occur. To predict these phenomena from a model-based perspective, faulted models are required for each defect type as in the case of mechanical components. Kral et al. [41] modelled an asymmetric induction motor (with the asymmetries arising from manufacturing flaws) using a model-based system design software. An induction motor with a broken rotor bar was simulated, and the electrical and mechanical performance was compared to experimental results. The model revealed that the fault was characterised by small sidebands to the fundamental wave current in the stator current, which were obscured by noise in the experiment. However, da Costa et al. [42] showed that these sidebands were visible in a purely experimental study, which suggests that signal processing is essential to analysing rotor faults. Stator fault models exist for turn faults [43] and inter-turn faults [44] revealed the challenges of modelling and detecting such faults in the stator current frequency spectrum is challenging. However, inter-turn stator fault detection has been successfully diagnosed using a model-based method on the time domain signal [45]. A model for a healthy motor was compared to the phase currents of a physical induction motor and the negative-sequence component of the estimation error was shown to be an accurate assessor of inter-turn short circuits and identified on which phase the fault occurs, and the severity of the fault.

Faults in the electrical motor feed such as unbalanced phase voltages or a grounded phase do not require a specific faulted model, but rather can be simulated using the appropriated inputs modelling the erroneous electrical signal for use as inputs to a nominal motor model. A common problem in induction motors is rather not electrical but mechanical: failed bearings. The modelling methodology associated to bearings has already been presented, but additional monitoring techniques are available for these faults via stator current measurements and motor efficiency calculations [46, 47, 48].

### Machine Learning Applications to Mechanical Components

Machine learning algorithms have been long popularized in domains of computer science and engineering, and are now gaining momentum in numerous fields such as finance [49, 50] and all branches of engineering and science [51, 52, 53, 54]. The significant interest in these algorithms is owed to their ability to solve problems that are either too demanding from a programming standpoint or impossible. In the application examined here by Siemens ISW, the effectiveness of teaching machine learning algorithms with model data to predict physical failure is investigated. In this section, the state of the art of machine learning techniques applied to rotating machinery is presented, with an emphasis on the most detection techniques for gear and bearing failures.

A key step in applying a machine learning methodology is the selection of features (inputs to the algorithm). Feature selection is particularly important in physical applications; certain physical failures may present detectable symptoms in idealized failure models that are obscured by noise or unfeasible to measure on a physical apparatus in an industrial application. The selected features must be resilient to noise and observable in non-stationary signals. It is unfeasible to train an ML algorithm to predict failure of a novel product using experimental data because one must produce this failure data using physical prototypes under accelerated-wear operating conditions. These conditions are not perfect analogues to true in-service loading conditions, and can be costly and time consuming processes. The state of the art of mechanical failure modelling of common drivetrain components and applications of machine learning to mechanical engineering is presented in this section, and this research acts as the base from which new developments of this project are created.

Limiting the analysis to the case of rotating machinery explicit restriction allows for the application of expert knowledge to intelligently select features in addition to mathematical selection criteria. In general, time series data itself is rarely used as a feature unless the ML algorithm is used to predict the dynamics of a system in a function analogous to a time integrator [55]. Signal processing techniques such as Fourier transforms [56], HHT transforms [57], discrete [58] or continuous [59] wavelet transforms and intrinsic mode functions [60, 61] are often applied before derived statistics such as spectral kurtosis, RMS, and other quantities [62, 63, 64] are used as machine learning features. Rauber et al. [65] developed a methodology to create and to select features for bearing fault diagnosis. A total of 130 features were created from times series statistical features, complex envelope analysis, and wavelet packet analysis and a subset of features was selected. With this intelligent method of feature selection, every classifier algorithm that was tested was able to classify bearing faults with a success rate over 95% for bearing faults of several sizes and several loads. As seen in Figure 13, the number of features is a critical parameter to investigate – using every feature results in a poor performance and significantly worse than using a single feature. In this case, 20 features results in optimal performance.

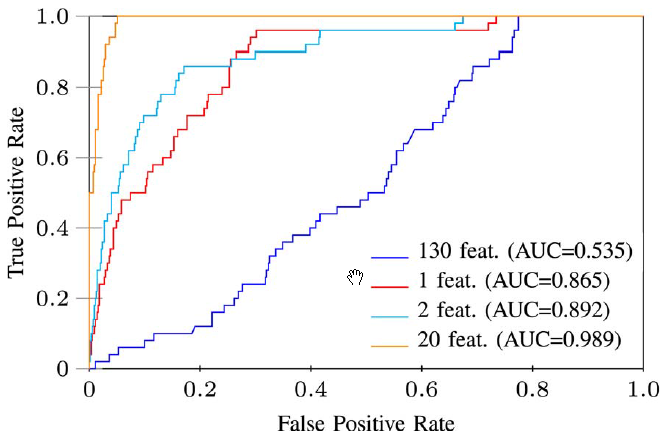


Figure 13. Classifier selection accuracy as a function of the number of features used. [65]

Machine learning algorithms are most often divided between classifiers and regression with applications to mechanical components limited in a large part to classification. The two most commonly used algorithms for this purpose regarding mechanical components are artificial neural networks (ANN) and support vector machines (SVM). ANNs have been applied to classify gear faults [66, 67], bearing faults [68, 69, 70], simultaneous bearing and gear faults [71], and planetary gearboxes [72]. However, they are well known to be able to describe complex nonlinear systems, and therefore one could expect their success in predicting failures on validation data produced using the same apparatus as the training data. They have also been used to predict dynamic responses in a smaller number of cases such as gear RMS behaviour [73] and to estimate the remaining useful life of bearings [74, 36, 75]. SVMs have been applied to classify bearing faults [76, 70], gear faults [66, 77] with success as well. In fact, with sufficient data, machine learning algorithm have proven effective at predicting failures across many devices such as induction motors [78, 79], wind turbine blades [80], and gas turbines [81]. Finally, sensor failure has been shown to be possible to detect using machine learning algorithms. Guo and Nurre [82] trained a neural network to detect “out of range” variables, and predict what the value should be based on the other variables. da Silva et al. [83] used model-based systems software to model bias, drift, scaling, and dropout sensor faults, as well as system faults for a solenoid actuator. The authors developed a fault correction system more complex than that of Guo and Nurre that could correct a signal based on its estimated type of failure. While this analysis is undoubtedly value to a system with many components, the priority in this section of the project is more restricted in scope, and sensor failures are a more general problem that have already been addressed.

These studies often lack direct application to many in-service mechanisms under varying conditions and loading histories, and the potential gains from applying the trained algorithms in these cases is unclear. In a select few cases, data was taken from publicly available databases or a large historical dataset and the results are likely to extend to in-service failures with greater fidelity. No literature was found in which a machine learning algorithm was trained on a model and validated using experimental data for mechanical fault detection.

* 1. Lessons Learned

Detecting physical faults in machines through intelligent data analysis and machine learning is the key focus of SISW in REFLEXION, and many insights into the possibilities and unique challenges were gained in the course of the project.

**Simulation-driven Machine Learning**

The most well-known applications of machine learning benefit from huge, human understandable, and widely available datasets such as images, audio and text for image recognition, speech-to-text, and machine translation. Applying machine learning in engineering applications faces far greater challenges on the topic of availability, access or the mere existence of training/testing data. Faults in rotating machinery most often require accelerometer measurements sampled in the 10-20 kilohertz range. Historically, this data was of little value (beyond monitoring RMS values for severe deviations) and was therefore not recorded. With advanced machine learning algorithms able to automatically detect faults in data of this nature, the data now has notable value but only exists for limited number of applications. Faults can take years or decades to develop for large, high-value machinery and numerous faults must occur to obtain a statistically relevant sample for algorithm training. Therefore, interim measures are required to enable machine learning solutions for fault detection without waiting for a representative number of failures to occur.

Simulation-generated data offers a potential bridging solution to enable condition monitoring before sufficient in-service data has been recorded. As their name implies, the utility of design models has been until now limited to the design phase of a product. Extending their applicability to the in-service phase leverages the significant effort and engineering knowledge incorporated in these models to enable OEMs or product users to better understand how their machines are operating or failing. Augmenting design models with faults is not, however, a trivial or straightforward process. The processes leading to common failures in rotating machinery such as bearing spalling or gear pitting/cracking are rooted in material fatigue, which is described by multiscale processes spanning orders of magnitude in length and time scale. Qualitative knowledge from historical failures is key in understanding the progression of faults from initiation of failure so that they can be represented in simulation models. Depending on the method of measurement, the transfer path between the fault site and the sensor must also be accounted for such that damping or dispersion is correctly modelled in the simulation-generated training data. Normalization to remove transfer path effects is a valid alternative if desired outcome from the machine learning algorithm does not rely on the magnitude of the measured values.

Within the context of research, simulation-driven machine learning is a challenging topic as it requires both design models and a physical test bench setup in which one can introduce (potentially irreversible) faults to create testing data. As a simulation and testing software/hardware provider, SISW possesses neither of these for industrial machines. However, SISW does possess a small benchtop drivetrain simulator into which faults including bearing faults can be added. Considering the significant interest and research into bearing fault detection, SISW selected this topic as a research focus for simulation-driven machine learning. This work will not be detailed here, except to note that simulation-generated data was successfully used to detect faults in physical bearing measurements with >90% accuracy, and the results were published [84] in the peer-reviewed journal “Mechanical Systems and Signal Processing.”

Purely simulation-driven studies in which both the training and testing data is simulation-generated has less value than experimentally-validated trained models, but can offer worthwhile insights for proofs of concept. Condition monitoring has existed for decades using thresholds on RMS values of acceleration or other simple statistics that can capture failures that are manifested in an increase in overall vibration. However, failures can appear in far more subtle ways in many components that cannot be detected through simple RMS values or even complex feature extraction using time/frequency/time-frequency methods. Deep learning proves to be a viable alternative to these methods, but creating a physical prototype and injecting failures is a costly process and expending this effort for a speculative experiment is unlikely. In this context, simulations are highly valuable as a source of labelled data. Machine learning algorithms for monitoring complex systems with highly nonlinear, discontinuous, or marginally observable phenomena can be proven (or disproven), the susceptibility to noise or measurement accuracy can be determined, and the necessary sensor set can be established. SISW researched applications of deep convolutional neural networks for fault detection and torque prediction for complex nonlinear drivetrains, which will be submitted for publication in a peer-reviewed scientific journal.

**Machine Learning in an Engineering Context**

Beyond challenges associated with acquiring in-service data for machine learning solution development, many other complexities arise that are often not present in typical machine learning applications in technology companies. Fault perceptibility in multichannel signals is one of these complexities. In common datatypes such as images, audio, or text, a human can label, convert to text, or translate without any specific technical competency. In an engineering context, it is possible that even a domain expert cannot identify the presence of a fault from a subset of measurements alone.

One can introduce a physical fault into a machine and collect vibration measurements with which to train a fault detection algorithm with supervised learning. If a given machine learning algorithm fails to identify the faults and the faults are too complex for a human domain expert to identify, isolating the cause of the failure is challenging. Possible reasons could include a poorly chosen algorithm or neural network architecture, unobservability of the fault for the chosen measurements, imperceptibility of the fault (low signal to noise ratio), or any number of other unknown possibilities. This situation is not a niche case and was encountered in two separate studies in the REFLEXION project. General solutions to resolve these challenges are unclear and further research is required.

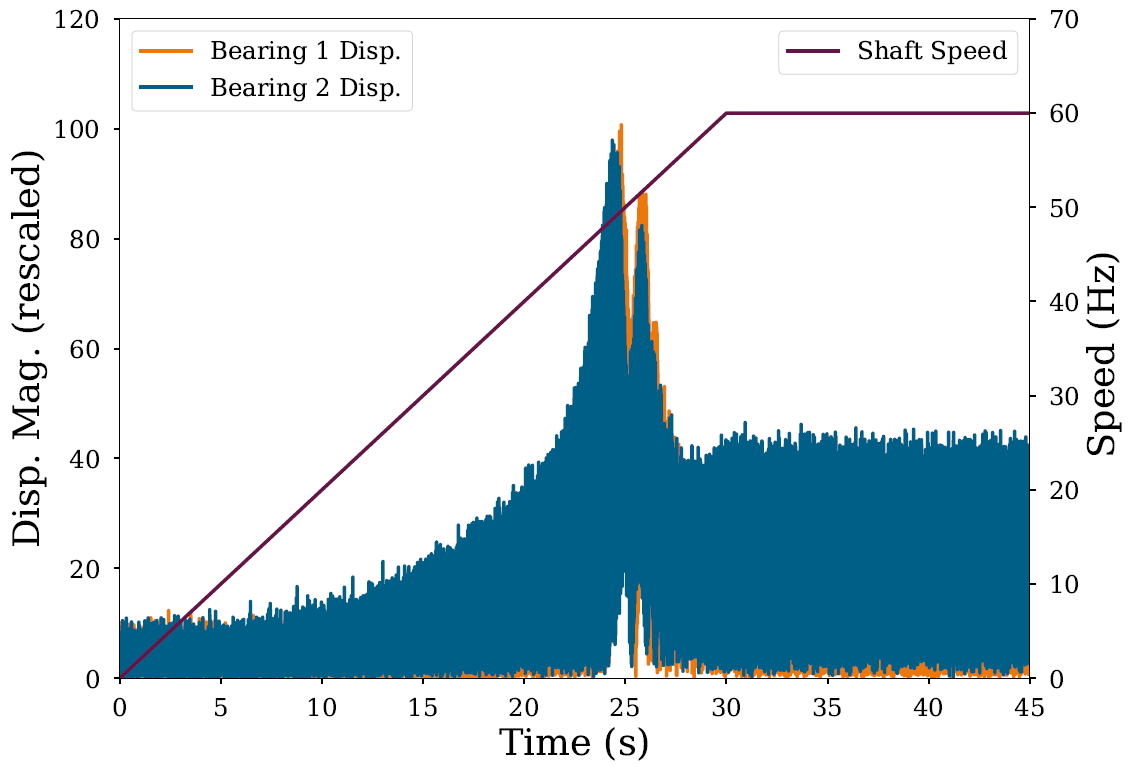


Figure 14. Magnitude of bearing displacements for an imbalanced shaft system. A resonance frequency is passed at 25s.

|  |  |
| --- | --- |
| Figure 15. Imbalanced shaft displacement at low speeds. Sensor noise obscures any indication of the imbalance. | Figure 16. The same shaft at higher speed shows clear oscillations at the shaft frequency. |

Challenges in neural network training related to these topics were first encountered with an application of convolutional neural networks to predict shaft imbalance parameters. A shaft with a static imbalance (such as a manufacturing defect resulting in a center of mass not aligned with the center of rotation) has a fault regardless of speed or operating conditions and a condition monitoring solution should accordingly predict the fault. Centrifugal force caused by the imbalance scales with the square of the rotational speed, thus the fault results in a higher signal-to-noise ratio as speed increases. Below some SNR threshold, the imbalance is imperceptible and yet the shaft is still imbalanced – in such a case, how should the data be labelled? The ideal answer is that the data should be labelled as faulty wherever the fault can be perceived, but determining this threshold of perceptibility is a challenge in practice. In this application, domain knowledge aids the process and an expert can set a speed threshold below which data is not used for training.

A variation on the notion of fault observation also arises in complex machines in which some components are not always active. A simple example is two springs in parallel, where one spring is shorter and interacts with the system only after the first has been partially compressed. If the second spring is cracked, this can only be detected if it is an active component in the system. Labelling the data as containing this fault for all time is erroneous and leads to total ineffectiveness of the machine learning model as it is trained to recognize nominal behaviour as faulty. Condition monitoring is possible in this scenario, but requires a sufficiently in-depth understanding of the physical device, and more importantly, the ability to discern when certain components are engaged. A simple real-world example is a cracked gear tooth; the crack only manifests change in the system when it is engaged and transmitting power, which is a small fraction of a revolution. In this case, monitoring can be easily achieved as the system has a fundamental periodicity (the synchronous speed of the shaft) and by examining over one or more full gear revolutions, the defect should be observable.

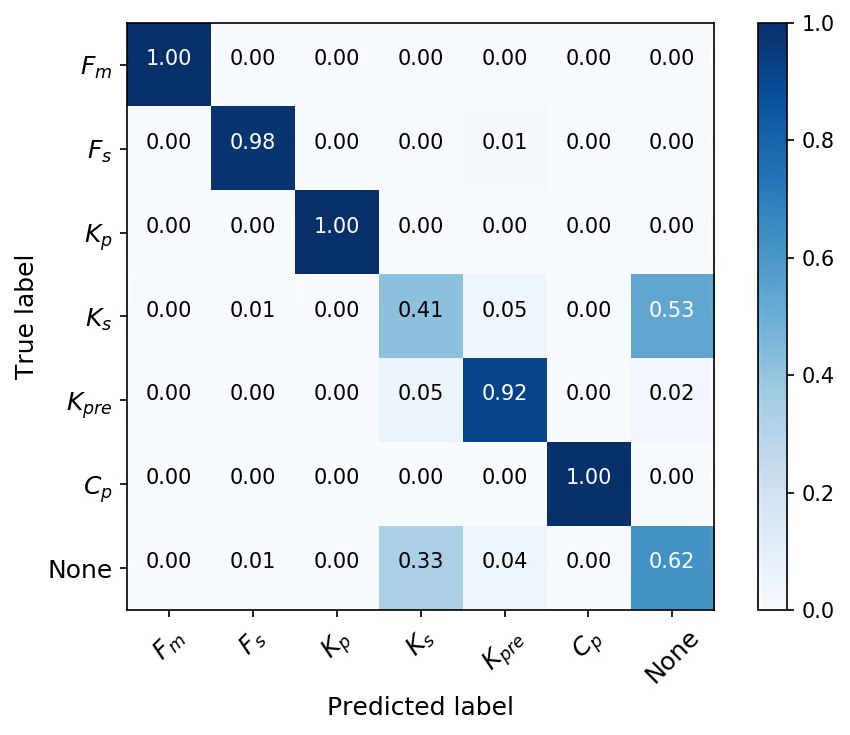


Figure 17. Confusion matrix for dual mass flywheel fault detection. The secondary spring fault Ks is often confused with no fault "None" because the second spring is only occasionally engaged.

The mechanism involved in the SISW study is a dual mass flywheel, which is used to damp driveline oscillations and is composed of two shafts linked by arc springs and friction plates (in highly simplified terms). While the mechanism rotates with a fundamental periodicity, the key oscillations are those in the relative angle of the shafts and this does not possess predictable periodicity (nor can this angle be predicted easily). Primary and secondary parallel arc springs transmit power between the shafts; however, the secondary spring is only activated with a sufficiently high relative angle and is only occasionally influencing the system. Figure 17 shows how the detection of a secondary spring fault is often confused with the no-fault case, which is caused entirely by this phenomenon. For systems without clear periodicity, at least with respect to possible faults and their interaction with the larger system, a bespoke analysis method is often required. An example of this is developing two machine learning models – the first to predict when a certain part is “active” or directly influencing the system, and a second, which is only applied when the first predicts the part is engaged, to perform condition monitoring.

**Fleet-based Condition Monitoring**

As previously indicated, it is often challenging to acquire large amounts of historical data (especially data containing fault occurrences). Besides the simulation-driven approach which was previously discussed, an alternative solution may be found through a fleet-based monitoring approach. In the fleet-based concept, “horizontal” comparisons (“machine-to-machine”) are made. The added value could for example be that anomalous behaviour can be more easily and more quickly detected, compared to the single machine case. Namely, under the assumption that the majority of the machines in the fleet are healthy, a deviating behaviour of a machine in the fleet can be indicative of a machine fault. Second, in case the machines are at the same location, this approach allows monitoring in a dynamic environment. While outlier analysis techniques like novelty detection can be used to avoid the need for an extensive historical dataset, they often fail in a variable environment as they assume that only a fault leads to novel data points. An unknown operational state or a change in the environmental factors might then be enough to trigger a false alarm. This is in contrast with a fleet-wide approach, as machines at the same location will respond similarly to environmental changes.

In order to understand the benefits and challenges with respect to a fleet-based approach, a prototype fleet setup was created and measured extensively (cf. D1.1). Research on the fleet dataset is currently underway, with first results reported in the CMMNO international conference [85]. This initial analysis already provided some valuable insights which are summarized into the following “lessons learned”:

* Even though the machines in the fleet may be nominally identical, there are always small differences which may complicate straightforward comparisons of the measured signals. For example, Figure 18 shows the phase currents which were measured at three drivetrains on the prototype fleet setup, under similar operating conditions (same operational speed of 820 RPM). Only drivetrain D1\_2 undergoes a phase unbalance fault during a limited duration. It is clear that there are differences in the magnitudes of the measured signals, which are equally large as the difference due to a phase unbalance fault. Conversely, it was found that (in this specific case), a comparison of the waveforms of the signals should allow for a more robust fault indicator, cf. Figure 19. As further detailed in [85], a non-standard measure based on Dynamic Time Warping (DTW), which focuses on the amount of warping, was successful for this particular problem. The generic applicability of the approach however needs to be further evaluated. Aside from this, a general lesson is that the features extracted from the data should always be critically reviewed. Ideally, features with low inter-machine variabilities but high sensitivities to faults are selected. In absence of operational data with fault occurrences, domain-specific expertise may be required for this purpose.

|  |  |
| --- | --- |
| Figure 18. Comparison of phase currents measured on three different drivetrains. Only D1\_2 (in green) has a phase unbalance fault for a limited duration. | Figure 19. Phase current waveform comparison faulty (red) vs. healthy (green) drivetrain. |

* During the measurements, the machines were also undergoing an identical run-up where the operational speed is gradually increased, hence presenting a more challenging example of condition monitoring in a nonstationary environment. The aforementioned DTW-based measure was calculated at different moments in time during the run-up, and used as input to an agglomerative hierarchical clustering approach. As shown in Figure 20, the method correctly identifies the 2 faulty drivetrains at lower and medium rotational speeds (under the constraint that every drivetrain is undergoing a nearly identical speed change). When applied at higher rotational speeds, this is however no longer the case, cf. Figure 21. The underlying physical reason is that from a certain speed limit, the inductance of the drivetrain becomes more important than the unbalance, such that comparing the waveforms is indeed not useable anymore for fault detection. As a general lesson, it can be stated that it is important to make use of domain knowledge in order to understand in which situations fleet-based comparisons can or cannot be made.

|  |  |  |
| --- | --- | --- |
|  | |  |
| Figure 20. Dendrogram showing hierarchical clustering results during run-up, at *low* instantaneous operational speed. (Ground truth: D1\_2 and D2\_10 are faulty) | Figure 21. Dendrogram showing hierarchical clustering results during run-up, at *high* instantaneous operational speed. (Ground truth: D1\_2 and D2\_10 are faulty) | |

1. Synerscope
   1. State of the Art

### Introduction

Visualization is a fundamental tool for human understanding that is based on human outstanding ability to understand visual objects. It has been used to communicate complex information, to understand complicated multivariate relationships, and to discover hidden facts in data. However, the notion of visualization as a scientific field was introduced by McCormic et al [86] just less than thirty years ago in 1987. They defined visualization as follows:

*Visualization is a method of computing. It transforms the symbolic into the geometric, enabling researchers to observe their simulations and computations. Visualization offers a method for seeing the unseen. It enriches the process of scientific discovery and fosters profound and unexpected insights.*

Visualization can facilitate discovery of hidden facts and solving problems by providing a structure to explore and analyse data. It also provides a means to communicate the results to diverse users. Visualization can be described as the process to transform data into information. Larking and Simon assess visualization as an accessible form of knowledge representation [87].

Today visualization goes beyond preparing simple graphs and pictures. It extends to geographical information systems, graphical user interfaces, multidimensional tables and graphs, 3D animation and virtual reality.

Although a good visualization eases deriving information from data, a poor one can misguide users into the wrong direction. Therefore, as it is indicated by Schumann and Muller [88] that any visualization should be expressive, effective and appropriate. It should show exactly what the data contains; nothing more and nothing less. It should be intuitively recognizable and interpretable. And it should be cost-effective; Time and resources that are used to achieve a visual representation should be assessed with respect to the value of result.

The key challenges for information visualization include creating visual representations that utilize human visual perceptual capability and enhance human information comprehension.

### Characterization of the visualization problem

As Aigner et al. [89] discuss to generate effective visual representations, raw data have to be transformed into image data in a data-dependent and task-specific manner. Conceptually, raw data have to be mapped to geometry and corresponding visual attributes like colour, position, size, or shape, also called visual variables [90] [91]. The visualization processes is determined by the data, the technique, and the interaction used. In the following section we will describe each of these elements.

#### Data

The central element of visualization is data. Data type, volume, velocity and veracity define suitable visualization techniques. These data features are set by the system that the data originates from and the desired visualization result depends on them.

Based on the pyramid framework by Mennis et al. [92] data can include three perspective levels: where (location), when (time) and what (theme). Where and when define domain of data and its dimensions. What describes what has been measured, observed or computed. To have a holistic view of the data all the three levels need to be considered in visualization.

Data can range through one, two or multi-dimensions. One dimensional data has one dense dimension. A typical example of one dimensional data is events occurring through time. Data that has two distinct dimensions are two dimensional data. Geographical data are a typical example of this type of data and maps are the typical visualization method. Any data set that has more than three attributes is a multi-dimensional data. Relational databases with multiple number of tables and tens to hundreds of columns are a typical example.

In addition to dimension data can range through different types. It can be number, text, graph, sensor log, picture, audio and video. In general these data types belong to two category of structured and unstructured data. Comparing these two categories from the visualization point of view, visualization of unstructured data has higher complexity and involves initial data mining processes to drive information and identify patterns from the data.

*Numbers* are the most common type of structured data and one of the first type that has been ever visualized. Since they already have a measurement scale it is easier to present them visually.

*Text* cannot be directly transformed to informative visualization. Since it is not easily describable by numbers, most of the standard visualization techniques cannot be applied for it. First step is to transform unstructured data to structured, derive patterns within the structured data and then visualization. An example for a simple transformation is word counting (see ThemeRiver in [93]), which is often combined with a principal component analysis or multidimensional scaling [94].

*Networks* are the generic method to represents data with interdependencies. Transportation network flow or email communications are two common example of this type. An overview of hierarchical information visualization techniques can be found in [95] , and an overview of web visualization techniques in [96].

If we look at data generated by different machines in industry that maps simultaneous processes, we can identify time as a common dimension. Additional data types are mostly numbers or texts in form of logs, and they have interdependencies as well as hierarchy. Later on, we will focus on different visualization techniques for data with these feature.

#### Visualization techniques

Visualization goal, besides data features, is the other factor that specifies techniques to be used. Ward et al [97] distinguish three main goals for a visual representation: explorative analysis; confirmative analysis; and presentation of analysis results.

The goal of explorative analysis is to get the initial insight into the data, to identify patterns, to extract holistic information and to generate hypotheses. Then these hypotheses are proved or disproved through examination and directed search in confirmative analysis phase. Last step is to present results of analysis and communicate findings with common readers [86].

Great variety of visualization techniques have been introduced for different types of data. Each technique has its own features and is strong to show one aspect of data. As log data are the focus of our project and we identified them as time-oriented data in section 8.3. we will introduce some visualization techniques that are more suitable for this type and give some examples.

#### Interaction and distortion techniques

Although techniques used is an important factor to get insight into data, alone it is not enough to achieve the three main visualization goals. To be able to explore the data and to confirm results, it is important to look into data from different perspectives and dimensions. In other words combine different visualization techniques together examine different hypothesis simultaneously. The way is a visualization platform in which it is possible to interactively filter or accentuate a subset of data.

A visualization is interactive if data analyst can directly interact with the visualization and depending on the objectives change the visualization dynamically. Interactivity lets users to investigate events and trends, to perform directed search, and to filter or accentuate a portion of data or, even zoom in to single data point.

An interactive visualization usually includes following features:

* Filtering; process of drilling down and focus in an interesting subset of data,
* Zooming; process of presenting more details on higher zoom levels,
* Distortion; process of exploring the data by preserving an overview during drill-down,
* Linking and brushing; process of inter-linking different visualizations together and automatic update of all when one is changed.

In general all these features are developed around the user’s intentions to interactively adjust visual representations to the tasks and data at hand. Yi et al [98] categorize users’ intentions into six *show me* prefaces groups: show me something else (explore); show me a different arrangement (reconfigure); show me a different representation (encode); show me more or less detail (abstract/elaborate); show me something conditionally (filter); show me related items (connect).

Figure 22 shows an example of a multi-view visualization created using the SynerScope Marcato platform [99]. All the views are inter-connected and selecting a subset in one view results in automatically change of the other views.

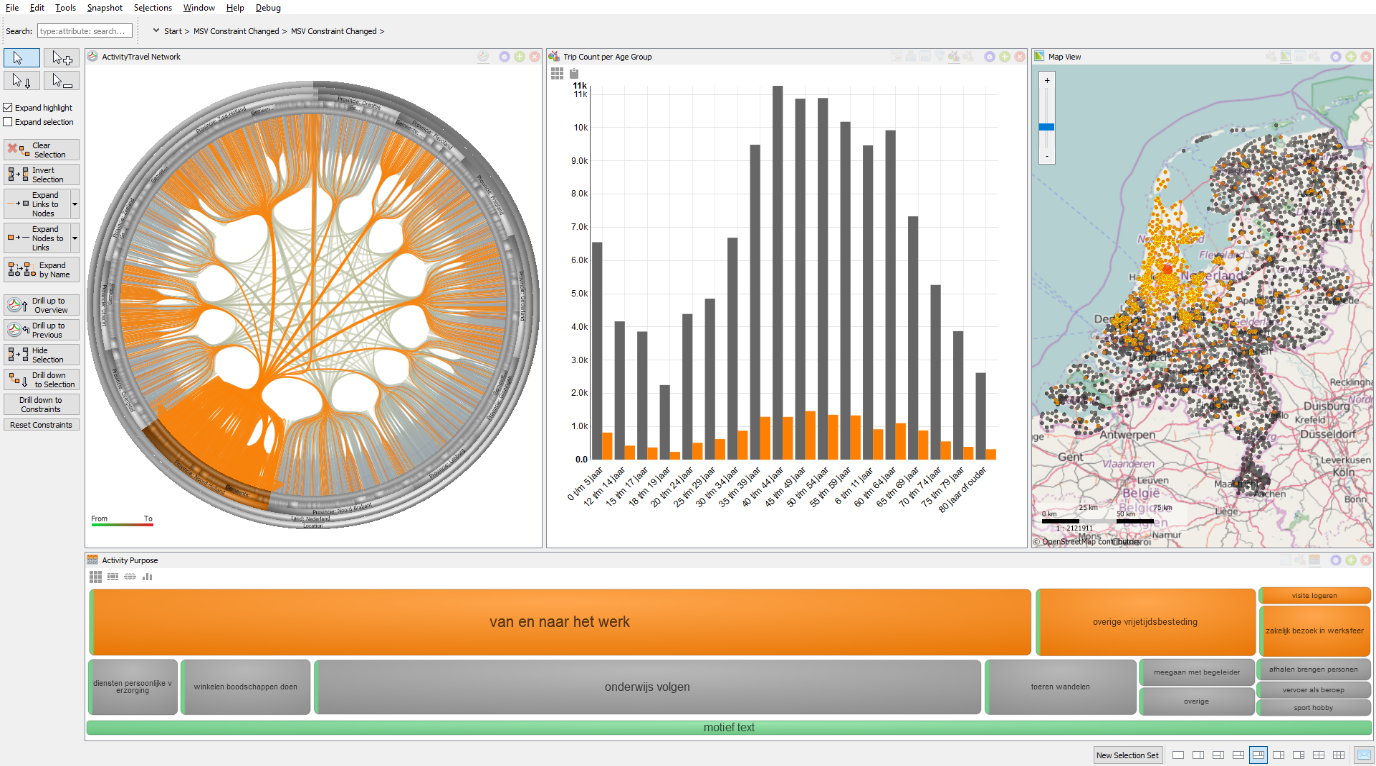


Figure 22. A multi-dimension, interconnected visualization created using SynerScope, Marcato platform.

### Visualization techniques for time-oriented data

In this section we introduce some of the visualization techniques that are developed to present time-oriented data.

#### Scatter plot

Scatter plot is mathematical diagram using Cartesian coordinate system to represent typically two dimensional representation of qualitative data on two axis. As it is described by [100] the data aspects are visualized by calculating the distance from the main axis. This is a straightforward technique to visualized time-oriented data.

#### Bar chart

Bar charts are commonly used charts where rectangular bars represent grouped data and lengths of these bars depict their value. Bar charts make comparison easy. However they just can be implemented for the variables with a ratio scale. Bar charts are also a good technique to represent information related to events in time such as number of event on a day, a week, and weekends.

#### Sparklines

Sparklines are type of visualization that you often find in web pages especially in stock market showing trends. As it is described by Tufte [101] they are simple, word-like graphics intended to be integrated into text. They are mainly used to provide extra information about a variable changes over time and they can be easily integrated into texts or tables.

#### TrendDisplay

Brodbeck and Girardin [102] introduce TrendDisplay technique to analyse trends in larger time-series. This technique includes two panels: main panel shows the raw data and the second panel shows derived statistical values. TrendDisplay can be used in quality control processes.

#### Timetree

There are different visualization techniques to present hierarchal organizational structures in a set of data. TimeTree by Card et al. [103] is an example in which element of time is represented using a time slider. The time slider enables users to navigate to any point in time.

#### Intrusion monitoring

Intrusion monitoring is a technique to visualize time-stamped network related log messages. Erbacher et al. [104] uses a central glyph to present the monitored system, radially arranged lines to show events and smaller glyphs at the end of the lines to represent remote hosts.

#### Tile maps

A tile map, presented by Mintz et al. [105], is a visual representation of a series of data as a calendar. For example, daily measurements are displayed in a matrix where each tile represents a day, a column represents a week and a row all values for a weekday. By varying the lightness of individual tiles values are shown.

#### EventViewer

EventViewer by Beard et al. [106] is a framework with three kinds of nested display elements called bands, stacks, and panels. Bands act as a container for a set of events where time is shown horizontally and bars within the band show events. Stacks are 'stacked' event bands and panels are groups of stacks. These display elements are organized on lattices or hierarchies along space, time and theme, forming small multiples.

* 1. Lessons Learned

Visualization component improvements

SynerScope Iximeer is a modular flexible visualization framework for the exploration and analysis of sensor (IoT) data. It is a generic platform to load structured time-series data comprising numbers, text, and geo-referenced points. Also, unstructured data such as video, images, and text are supported. Users can explore this mix of structured and unstructured data in context using a flexible setup of different visualizations that are automatically linked. This link navigation supports fast hypothesis building for visual validation. Current modules include scatterplot, word-wall, bar-chart, histogram, density map, and raw data view.

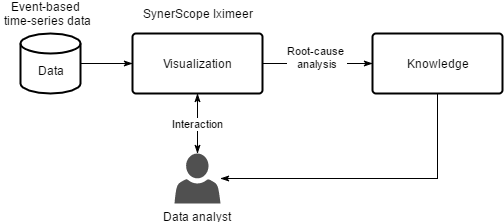


Figure 23. Root-cause analysis on event-based time-series data.

During the investigation of IoT sensor-based data we used the standard visualizations available in the SynerScope solution. It proved very useful that visualizations are automatically linked such that items of interest can be viewed from different perspectives. The visualizations used in the exploration and analysis setup are scatterplots, bar-charts, histogram, a map and a word-wall. However, due to the large number of different attributes involved, the available visualizations are deemed not enough for a holistic investigation.

The investigations created are often in the context of root cause analysis (see Figure 23) of some device. In a typical IoT sensor-based data stream there are > 1000 variables involved. All these variables are captured because it is beforehand not clear what variables contain a pattern or clue that leads to the identification of the root-cause. Typically, in the visualizations available only up to about 10 variables can be shown effectively. From the use cases it was deemed important by the domain experts that more variables are shown in a plot. For this a parallel coordinate plot (see Figure 24) could be used. In a typical parallel coordinate plot the different variables are plotted as horizontally aligned vertical axes that represent the ranges of the according attribute. For each data item a line is drawn that crosses each axis at the position of the data value. If performed in a naive way this introduces overdrawing problems. Therefore some density based techniques should be used. Furthermore, if many attributes are involved this possess challenges on the visualization, due to limited horizontal screen space. A solution to this may be to show only a subset selection of the attributes. For this mechanism to be effective some ranking should be computed on the attributes for example based on ranges, correlation, or mean and standard deviations. Also, the ordering of the axis becomes important as not all combinations of attributes can be shown. Again, here correlation-based approaches that identify positive and negative correlations and renders these axis early on, or just shows these in isolation and leaves out the rest of the axes. This implies that axis with no correlation are less important or interesting.

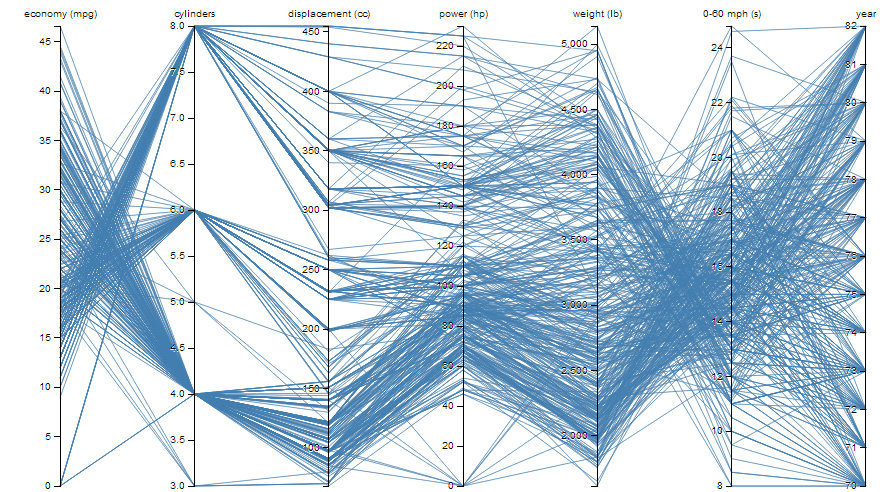
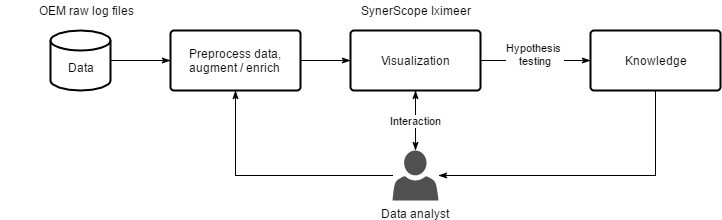


Figure 24. Parallel coordinate plot showing 7 data variables simultaneously.

Event streams



**Figure 25 - Analysis and exploration of IoT data event stream.**

We identified that for the analysis and exploration of IoT data event streams are a necessity. Often the data is not in a nice format and we need a method to transform event data to grouped event streams to unlock the data for analytics (visualization, data mining and anomaly detection).

The first step to grouped event stream analytics (see Figure 25) is the need for a closer interaction with SQL databases that store large event data. We have implemented direct connectivity between Ixiwa, Iximeer and Postgres in order to streamline the analysis process. Furthermore, the linkages between datasets should be easy to identify to create the necessary context. We found that often non-correlated data can be quite important for building the context to understand the scope of a problem during analysis. For this, the bottleneck in the process was locating and identifying the right data sources. We have expanded our Ixiwa product to help identify correlations between data sources, and have successfully used this to steer the user during the data discovery phase, shortening the time of analysis. The key differentiating capability has been to quickly ingest large data collections, and sift through the data, using these similarities as an accelerator.

These features where identified from interactions with partners in the project. We have evaluated the features of our Ixiwa and Iximeer products and these have led to the above described additions.

The addition of structured log analysis by means of grouped event streams has opened up initial customers for SynerScope in Energy and Oil&Gas.

1. Yazzoom
   1. State of the Art

The field of predictive analytics is broad in many respects: there is a wide range of application areas, algorithms, implementations and problem settings. The majority of data analytics, both in a business and industrial context, still consist of so-called descriptive analytics (dashboarding, reporting, ad-hoc or on-demand analytics etc.) whereas the more advanced applications involve building predictive analytics models. However, not all use cases can be cast into the form of a predictive analytics problem (whereby a clear goal or variable to be predicted can be defined). Rather, many companies have the more general need to monitor the general condition and state of their processes, systems or business and to be able to rapidly respond to *anomalous* events, to do *a posteriori* investigation of certain incidents or in general to assess any deviations from normal behavior. Stated otherwise: very often companies are interested in anomalies in the data, without being able to or wanting to specify what anomalies are beforehand.

Anomaly detection consists in identifying patterns in data that do not conform to a well- defined ‘normal’ behavior. Anomalies can be such because of the value of a single instance alone (*point anomalies*), in relation to some neighboring context (*contextual anomalies*) or the values of multiple subsequent instances only when considered together (*collective anomalies*) [107].

Anomaly detection today has a wide gamut of use cases: wireless sensor networks, where particular attention is posed to distributed, energy-efficient and lightweight solutions [108, 109]; network intrusion detection, with the focus of protecting target systems against malicious attacks and intrusions [107]; fraud detection [110]; industrial machines faults and damages [111, 112] and many others. In all its applications, many challenges have to be faced, such as the difficulty to model the distinction between *‘anomaly’* and (unwanted) noise; adapting the model to dynamic changes in the *‘normal’* behavior; providing early detections while keeping the number of false alarms to a minimum.

In general, this is a challenging task, considering it is hard to precisely and conclusively define and model a ‘normal’ behavior, and how this definition varies depending on the context. Moreover, when data comes from a large number of heterogeneous sources, forming large datasets (Big Data), it becomes computationally hard and expensive to mine them looking for anomalies, usually amounting only for a very small percentage. In this context, standard processing techniques based on Business Intelligence rules are not a viable option anymore, and new methods and trends are being explored to perform analysis and scale to the size of current datasets. Many different methods coming from several different research areas such as Machine Learning, Statistics, Information Theory and more, have already been proposed to deal with this problem [113].

### Anomaly detection in machine learning

Anomaly detection in machine learning can be defined as a *supervised*, *unsupervised* or *semi-supervised learning* task depending on the available data:

* in a *supervised* approach, a classifier is trained on part the available data (*training set*) consisting of labelled data both for the normal (or “*positive*”) data and for the anomalies (“*negative*”), and evaluated against an unseen set (i.e. not used for training), called *test data*;
* in the *semi-supervised* version, the training set only contains labelled *positive* data, without any labeled anomaly; in this case, the approach is to train the classifier to model the *normal* behaviour, and then evaluate it against the test data to identify possible anomalies that don’t conform to (are not generated by) such model;
* the *unsupervised* approach does not require any of the data to be labelled, and automatically builds a model to distinguish between anomaly and normal behaviour, under the assumption that positive data is much more common than anomalies.



Figure 26. Different settings for anomaly detection, depending on the availability and usage of labelled data [114]

In all the applications where it is challenging to label data or where it is not known an anomaly can be defined, the *unsupervised* approach is the preferred option. The output of an anomaly detection system can be either a *label* (or *class*) in the supervised and semi-supervised versions, i.e. the system classifies each instance as anomalous/normal, or a *score,* in the unsupervised approach, quantifying how anomalous each instance is according to a chosen scoring criterion (e.g. distance from nearest neighbors, distance from the cluster centroid etc.). This then requires an additional step where some threshold for the score is set, to consider an instance as anomalous, and often this definition can be nontrivial. With the new ubiquitous and pervasive technological systems monitored by sensors (e.g. IoT), more and more data is measured and stored every second, leading to one of the major challenge faced by anomaly detection methods nowadays: dealing and scaling algorithms and methods to the size of huge heterogeneous datasets. In addition to the dimensionality problem, another issue arises when irrelevant attributes contained in such high-dimensional data could have a negative impact on anomalies, by hiding or modifying properties critical for the detection [115].

Several families of methods exist for anomaly detection, and it is possible to classify them according to the machine learning algorithm they are based on:

* **Methods based on classifiers**: these techniques use known machine learning algorithms to tackle the problem as a standard classification problem, with some specific anomaly detection related challenges, such as the data imbalance (usually anomalies account for a very low percentage of the whole dataset). Detection system(s) using algorithms like Neural Networks, Bayesian Networks and SVM have been reported in literature. This group of techniques requires labeled data so it is not generally applicable for unsupervised problems [113].
* **Methods based on Nearest-Neighbors (NN)**: these methods operate under the assumption that ‘normal’ observations lie within a dense neighborhood while anomalies are located far from their nearest neighbor, according to a well-defined measure of distance between observations. In its simplest form, the k-NN technique computes the distances between pairs of *k* neighboring instances and identifies as anomalies those instances with the highest distance from the neighbors. These techniques are usually used in unsupervised settings, but there are also examples of applications for semi-supervised problems. Some modifications of it, such as LOF (Local Outlier Factor) approach, extend the technique to reveal also local anomalies. LOF computes the *local density* of every instance and identifies as *local* outliers those instances with a significantly lower density than that of the neighbors. In general, this approach has several limitations, including the polynomial computational complexity, the sensibility to parameter changes (*k*, scoring threshold) and the inability to find local outliers.
* **Methods based on clustering**: these techniques aim to cluster observations into several groups and detect anomalies either as (1) *observations* that don’t belong to any cluster, or (2) *observations* that are far from their cluster’s centroid or (3) *observations* that belong to small or sparse clusters, according to the chosen strategy. The anomaly score is based on the distance to the cluster centroid. The primary application of these techniques is for unsupervised or semi-supervised settings.
* **Methods based on statistics**: these techniques operate building a stochastic model of the data and consider anomalies as those points with low probability of being generated by the model. To determine whether an unseen observation is generated by the model, statistical inference is employed. These methods can be *parametric*, when the parameters of the model can be estimated from the data itself since the underlying distribution of the data is assumed known, or *nonparametric*, when such assumption cannot be made.

While these techniques refer to point anomaly detection, there exist ways to adapt them for **contextual anomaly detection**. Depending on the data, context can be:

* the *spatial* neighborhood of an instance;
* the set of neighbors of a node for *graph-based data*;
* the position in a sequence for *sequential* data (e.g. time series);
* the *profile* of an instance, for data segmented or clustered into groups (*profiles*).

So far, the research on contextual anomaly detection methods has been limited. Broadly, it is possible to categorize the proposed techniques in two approaches:

1. Reducing the problem to point anomaly detection:
   1. the first step consists in identifying the context of an instance using the contextual attributes;
   2. then employing a point anomaly detection technique on the instance within its context to compute the score.
2. Modeling the structure of the data and employing it to detect anomalies:
   1. It is not always easy to segment the data into contexts of every instance, especially in time-series and sequential data. The first step here is to build a model of the training data to predict the expected behavior w.r.t. a given context - the actual behavior is then compared to the expected behavior and an anomaly is raised whenever the difference is significant.

### Anomaly detection on time series data

Anomaly or outlier detection on time series data is an active research field in itself, which has been studied in the context of a large number of application domains, including financial markets, system diagnosis, biological data and user-action sequences [116]. Correspondingly, depending on the precise context, settings and nature of the problem of each of these fields, a variety of different methods have been proposed. Depending on the application area and type of anomaly one wishes to detect, the type of outlier detection can be classified as either outliers in a time series database or finding outliers within a given time series, which can be further subdivided into identifying points or subsequences as outliers. The specific context of dealing with streaming data as opposed to offline anomaly detection on time series poses its own challenges and solutions. Methods used here include Dynamic Bayesian Networks [117], local outlier factor detection [118]. For the case of high-dimensional streaming data, SPOT (Stream Projected Outlier deTector) is proposed [119].

### Anomaly detection for log analysis

Information about complex systems of interacting components can usually be obtained from online continuous measurements using different sensor types, or by examining log files generated by the individual components. Depending on various properties of these components (computational resources, power efficiency, storage limitations, …) these text logs can span a wide range in verbosity. Moreover, in complex systems there exist usually many different types of components which all have their own log types (server logs, router logs, event logs, …). These logs can in addition contain both numerical and non-numerical data. It is clear that generally speaking the extraction of relevant information suitable for processing by machine learning algorithms is a challenging task.

Several approaches to this problem have already been described in literature. Some methods approach the problem of problem diagnosis by encoding the system activity obtained from log analysis as a symbol string and apply analysis methods on these sets of strings [120, 121, 122]. Other algorithms view system logs as a series of footprints of execution [123, 124, 125]. Still other approaches try to detect more general problems (e.g. system misconfigurations [126]) or view it as a supervised learning problem and train decision trees when labelled examples are available [127].

Since logs are mostly textual data, several approaches have used techniques from natural language processing to help extract information from the raw logs. An often used initial step is extracting N-gram (a continuous sequence of N tokens) frequencies from the logs [128]. These N-gram distributions are sometimes further processed to obtain TF-IDF (term frequency – inverse document frequency) scores for certain terms [129]: this metric quantifies the relative occurrence (and, it is assumed, importance) of a token in a document relative to its total occurrence in all documents in the dataset. Several approaches then apply some form of unsupervised clustering method to the extracted features, such as K-Means [130] or DBSCAN [131].

* 1. Lessons Learned

Throughout REFLEXION, Yazzoom has further developed its platform for anomaly detection on machine-generated data called Yanomaly. This platform, consisting of a backend made from several scalable software components communicating via a Kafka message queue and a frontend that allows configuration and deployment of models on various data sources, is built around a library of anomaly detection algorithms suitable for processing various types of data, namely combinations of univariate and multivariate, stationary or timeseries data, and text log files.

The major lessons learned (or further confirmed) from the participation in and collaboration with REFLEXION partners can be summarized as follows:

* Despite some levels of standardization in tools and data collection platforms (e.g. for log collection and analysis, the ELK[[5]](#footnote-5) stack is a de facto standard), there is still a large variety and diversity in log and numeric data formats. This is no surprise: on the one hand these formats are dictated by technical constraints (data bandwidth, sampling frequencies, heterogeneity), use case (troubleshooting, design, monitoring, …) and on the other hand by the legacy code already in place. This makes standardization (if at all possible) a slow process. For that reason, our lesson learned is to remain as flexible as possible w.r.t. the data format and not impose assumptions or requirements about the format of the data but rather enable the analytics methods to be able to deal with the variety of data occurring in the field. For instance, rather than limiting our log analysis pipeline to a subset of fixed log formats, we built an automatic log parser that learns the format of the logs from a sample set of data.
* It is a well-known (and theoretically corroborated) statement that there is no single predictive analytics algorithm that is generally usable for all types of data. This is also, and perhaps more so, the case for anomaly detection since often there is no ground truth available in those cases. For that reason, multiple different algorithms, each with their respective strengths and application domains, are necessary and were included in Yanomaly.
* Because of the somewhat ill-defined nature of anomaly detection problems, there is always some form of tuning of parameters involved which cannot be automated. Even though all methods are unsupervised in the sense that they do not require labelled anomaly instances, most of them do require some form of training data for which the only assumption is that it contains mostly ‘normal’ data points. However, depending on the final usage of the anomaly detection algorithm once it is trained, at the very least the detection threshold (the threshold value applied to the anomaly score, which then indicates an anomaly event) should be tuned to the use case – for instance, for the case of monitoring, the threshold is usually placed higher to avoid alarm fatigue than the case of troubleshooting, where the user is actively investigating the data in search of the root cause for a certain known incident. In addition to this parameter, usually some high-level parameters (so-called hyperparameters because the learning algorithms compute their internal parameters based on training data, whereas the hyperparameters affect the learning process itself) such as timescale parameters also need to be tuned. This can then be done using either or a combination of domain knowledge or in case it is available, some known labelled instances of anomalies.

1. Commonalities

Across the industries common themes can be identified.

* 1. Data collection and storage

Data collection and storage is in place. Log data is acquired and regularly uploaded to a central data storage where further processing is performed.

* 1. Data content

Very often, the data that is captured from the machines turns out to be incomplete or inappropriate to sufficiently answer the questions at hand. This is due to a combination of reasons:

* Data logging is not the primary concern of product development. Functionality and time-to-market are more important. As a consequence, the data content is not optimal.
* Determining upfront what data is required is difficult. Only after deployment problems surface and the need for additional data arises.

Improving data logging has very long lead times as deploying new versions might take years to the field, if at all possible in all cases. Some customers might refuse updates as they do not have immediate benefit from updating a well-performing machine, or changing a machine, even if it only concerns data logging capabilities, requires recertification.

As a consequence, extracting useful information from the data at hand requires a lot of thought and work, and even then quality of information is lower than one would hope for.

* 1. Data context

Interpreting the data is cumbersome as the operating environment (context) of machines is very important to understand the data, but this context information is not available because it is not known or prohibited to share because of legal or commercial reasons. The context has to be inferred from the data itself, leading to larger uncertainties in the conclusions. Alternatively, if expert knowledge is available, these uncertainties could be reduced.

* 1. Data quantity

Multiple megabytes or even gigabytes of data are produced per machine, per day. Although the data is abundant, *interesting* data, such as error situations, is not. This makes traditional machine learning applications difficult and requires dedicated, domain specific solutions.

* 1. Data versioning

Because machine versions change over time, and different versions are operational at the same moment in time, the data that is collected also differs over time. This makes data analysis over longer periods difficult.

In practice, this problem is circumvented by either focussing on a single machine release, taking the loss of more data for granted, or by ad-hoc data repair actions based on the knowledge of different versions. The need for data governance and model governance is recognised but a solution is not in place.

* 1. Data usage

The introduction of data collection and storage is typically initiated by the service organisations. These organisations can show the business case in which the benefits outscore the set-up costs.

When the data is made available, other divisions take advantage of it by using it for a wide range of purposes.

* 1. Dashboards

A trend that appears across the industries is to display the results of data analysis in operational dashboards in near real-time. Using advanced data analysis behind the scenes and smart visualisations, customers and/or manufacturers get live insight in the operational performance of their machines.

1. References & citations

|  |  |
| --- | --- |
| [1] | J. Tretmans, “Test Generation with Inputs, Outputs and Repetitive Quiescence,” *Software - Concepts and Tools,* vol. 17, no. 3, pp. 103-120, 1996. |
| [2] | E. Moore, “Gedanken-experiments on Sequential Machines,” in *Volume 34 of Annals of Mathematical Studies*, Princeton, N.J., Princeton University Press, 1956, p. 129–153. |
| [3] | M. Broy, B. Jonsson, J.-P. Katoen, M. Leucker and A. Pretschner, “Model-Based Testing of Reactive Systems,” in *Advanced Lectures*, Spring, 2005. |
| [4] | B. K. N. Rao, Handbook of Condition Monitoring, Oxford, UK: Elsevier, 1996. |
| [5] | R. B. Randall, Vibration-base condition monitoring: industrial, aerospace, and automotive applications, Jon Wiley & Sons, 2011. |
| [6] | L. Wan, M. Zeiler, S. Zhang, Y. L. Cun and R. Fergus, “Regularization of neural networks using dropconnect,” in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013. |
| [7] | G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath and others, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *IEEE Signal Processing Magazine,* vol. 29, no. 6, pp. 82-97, 2012. |
| [8] | D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot and others, “Mastering the game of Go with deep neural networks and tree search,” *Nature,* vol. 529, no. 7587, pp. 484-489, 2016. |
| [9] | M. Mohandes, T. Halawani, S. Rehman and A. A. Hussain, “Support vector machines for wind speed prediction,” *Renewable Energy,* vol. 29, no. 6, pp. 939-947, 2004. |
| [10] | N. Laouti, N. Sheibat-Othman and S. Othman, “Support vector machines for fault detection in wind turbines,” in *Proceedings of IFAC world congress*, 2011. |
| [11] | J. Lee, H. Kao and Y. Shanhu, “Service innovation and smart analytics for industry 4.0 and big data environment,” in *Prodedia CIRP*, 2014. |
| [12] | D. Siegel, Prognostics and Health Assessment of a MUlti-Regime System using a Residual Clustering Health Monitoring Approach, Cincinnati: University of Cincinnati, 2013. |
| [13] | B. e. a. Mathews, “Discovering Anomalous Aviation Safety Events Using Scalable Data Mining Algorithms,” *Journal of Aerospace Information Systems,* vol. 10, no. 10, 2013. |
| [14] | A. Parlos, K. Kim and R. Bharadwaj, “Sensorless detection of mechanical faults in electromechanical systems,” *Mechatronics,* vol. 14, no. 357-380, 2004. |
| [15] | A. e. a. Zaher, “Online wind turbine fault detection through automated SCADA data analysis,” *Wind Energy,* vol. 12, no. 6, pp. 574-593, 2009. |
| [16] | P. Johnson, “Fleet wide asset monitoring: Sensory Data to Signal Processing to Prognostics,” in *Proceedings of the Annual Conference of the Prognostics and Health Management Society*, 2012. |
| [17] | P. Paris and F. Erdogan, “A critical analysis of crack propagation laws,” *Journal of basic engineering,* pp. 528-533, 1963. |
| [18] | H. N. Ozguven and D. R. Houser, “Mathematical models used in gear dynamics - a review,” *Journal of sound and vibration,* pp. 383-411, 1988. |
| [19] | D. G. Lewicki and R. Ballarini, “Effect of rim thickness on gear crack propagation path,” *Journal of mechanical design,* pp. 88-95, 1997. |
| [20] | D. Yang and J. Lin, “Hertzian damping, tooth friction and bending elasticity in gear impact dynamics,” *Journal of mechanisms, transmissions, and automation in design,* pp. 189-196, 1987. |
| [21] | S. Wu, Ming, Z. J. and A. Parey, “Simulation of spur gear dynamics and estimation of fault growth,” *Journal of Sound and Vibration,* pp. 608-624, 2008. |
| [22] | Z. Chen and Y. Shao, “Dynamic simulation of spur gear with tooth root crack propagating,” *Engineering Failure Analysis,* pp. 2149-2164, 2011. |
| [23] | Z. Wan, H. Cao, Y. Zi, W. He and Z. He, “An improved time-varying mesh stiffness algorithm and dynamic modeling of gear-rotor system with tooth root crack,” *Engineering Failure Analysis,* pp. 157-177, 2014. |
| [24] | S. Jia and I. Howard, “Comparison of localised spalling and crack damage from dynamic modelling of spur gear vibrations,” *Mechanical Systems and Signal Processing,* pp. 332-349, 2006. |
| [25] | J. Wang, Numerical and experimental anlysis of spur gears in mesh, 2003. |
| [26] | H. Ma, X. Pang, R. Feng, R. Song and B. Wen, “Fault features analysis of cracked gear considering the effects of the extended tooth contact,” *Engineering Failure Analysis,* pp. 105-120, 2015. |
| [27] | H. Ma, X. Pang, J. Zeng, Q. Wang and B. Wen, “Effects of gear crack propagation paths on vibration responses of the perforated gear system,” *Mechanical Systems and Signal Processing,* pp. 113-128, 2015. |
| [28] | J. Flasker, G. Fajdiga, S. Glodez and T. K. Hellen, “Numerical simulation of surface pitting due to contact loading,” *International journal of fatigue,* vol. 23, no. 7, pp. 599-605, 2001. |
| [29] | J. Jurenka and M. Spaniel, “Advanced FE model for simulation of pitting crack growth,” *Advances in Engineering Software,* vol. 72, pp. 218-225, 2014. |
| [30] | A. F. del Rincon, F. Viadero, M. Iglesias, A. de-Juan, P. Garcia and R. Sancibrian, “Effect of cracks and pitting defects on gear meshing,” *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science,* pp. 2805-2815, 2012. |
| [31] | F. Chaari, R. Zimroz, W. Bartelmus, T. Fakhfakh and M. Haddar, “Modeling of local damages in spur gears and effects on dynamics response in presence of varying load conditions,” *Proceedings of Surveillance,* vol. 6, pp. 1-19, 2011. |
| [32] | Y. Guo and R. G. Parker, “Purely rotational model and vibration modes of compound planetary gears,” *Mechanism and Machine Theory,* pp. 365-377, 2010. |
| [33] | Z. Feng and M. J. Zuo, “Vibration signal models for fault diagnosis of planetary gearboxes,” *Journal of Sound and Vibration,* pp. 4919-4939, 2012. |
| [34] | M. Inalpolat and A. Kahraman, “A theoretical and experimental investigation of modulation sidebands of planetary gear sets,” *Journal of Sound and Vibration,* pp. 677-696, 2009. |
| [35] | J. FRE and T. CICONE, Friction in Lubricated Contacts, 2001. |
| [36] | Y. Shao, J. Liu and J. Ye, “A new method to model a localized surface defect in a cylindrical roller-bearing dynamic simulation,” *Proceedings of the Institution of Mechanical Engineers, Part J: Journal of Engineering Tribology,* pp. 140-159, 2014. |
| [37] | P. a. S. J. McFadden, “Model for the vibration produced by a single point defect in a rolling element bearing,” *Journal of sound and vibration,* pp. 69-82, 1984. |
| [38] | D. Brie, “Modelling of the spalled rolling element bearing vibration signal: an overview and some new results,” *Mechanical Systems and Signal Processing,* pp. 353-369, 2000. |
| [39] | S. Sassi, B. Badri and M. Thomas, “A numerical model to predict damaged bearing vibrations,” *Journal of Vibration and Control,* pp. 1603-1628, 2007. |
| [40] | M. a. B. M. Tadina, “Improved model of a ball bearing for the simulation of vibration signals due to faults during run-up,” *Journal of sound and vibration,* vol. 330, no. 17, pp. 4287-4301, 2011. |
| [41] | C. a. H. A. a. G. C. Kral, “Modeling and Simulation of Broken Rotor Bars in Squirrel Cage Induction Machines,” in *Proceedings of the World congress on engineering*, 2009. |
| [42] | C. da Costa, M. Kashiwagi and M. H. Mathias, “Rotor failure detection of induction motors by wavelet transform and Fourier transform in non-stationary condition,” *Case Studies in Mechanical Systems and Signal Processing,* vol. 1, pp. 15-26, 2015. |
| [43] | R. M. Tallam, T. G. Habetler and R. G. Harley, “Transient model for induction machines with stator winding turn faults,” in *Conference Record of the 2000 IEEE Industry Applications Conference, 2000.* , 2000. |
| [44] | G. M. a. P. J. Joksimovic, “The detection of inter-turn short circuits in the stator windings of operating motors,” *IEEE Transactions on Industrial Electronics,* vol. 47, no. 5, pp. 1078-1084, 2000. |
| [45] | C. H. De Angelo, G. R. Bossio, S. J. Giaccone, M. I. Valla, J. A. Solsona and G. O. Garcia, “Online model-based stator-fault detection and identification in induction motors,” *IEEE Transactions on Industrial Electronics,* vol. 56, no. 11, pp. 4671-4680, 2009. |
| [46] | L. Frosini and E. Bassi, “Stator current and motor efficiency as indicators for different types of bearing faults in induction motors,” *IEEE Transactions on Industrial Electronics,* vol. 57, no. 1, pp. 244-251, 2010. |
| [47] | R. R. Schoen, T. G. Habetler, F. Kamran and R. Bartfield, “Motor bearing damage detection using stator current monitoring,” *IEEE Transactions on Industry Applications,* vol. 31, no. 6, pp. 1274-1279, 1995. |
| [48] | Y. Amirat, V. Choqueuse and M. Benbouzid, “EEMD-based wind turbine bearing failure detection using the generator stator current homopolar component,” *Mechanical Systems and Signal Processing,* vol. 41, no. 1, pp. 667-678, 2013. |
| [49] | L. Gyorfi, G. Ottucsak and H. Walk, Machine learning for financial engineering, World Scientific, 2012. |
| [50] | C. Carter and J. Catlett, “Assessing credit card applications using machine learning,” *IEEE expert,* pp. 71-79, 1987. |
| [51] | M. A. Shipp, K. N. Ross, P. Tamayo, A. P. Weng, J. L. Kutok, R. C. Aguiar, M. Gaasenbeek, M. Angelo, M. Reich, G. S. Pinkus and others, “Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning,” *Nature medicine,* pp. 68-74, 2002. |
| [52] | K. Rajan, “Materials informatics,” *Materials Today,* pp. 38-45, 2005. |
| [53] | H. a. C. J. A. a. M. Y. L. a. C. M. Guo, “Automotive signal diagnostics using wavelets and machine learning,” *IEEE Transactions on Vehicular Technology,* vol. 49, no. 5, pp. 1650-1662, 2000. |
| [54] | I. Kononenko, “Machine learning for medical diagnosis: history, state of the art and perspective,” *Artificial Intelligence in medicine,* vol. 23, no. 1, pp. 89-109, 2001. |
| [55] | S. Chen and S. A. Billings, “Neural networks for nonlinear dynamic system modelling and identification,” *International journal of control,* pp. 319-346, 1992. |
| [56] | D. Tax, A. Ypma and R. Duin, “Support vector data description applied to machine vibration analysis,” in *Proc. 5th Annual Conference of the Advanced School for Computing and Imaging (Heijen, NL)*, 1999, pp. 398-405. |
| [57] | J. Cheng, D. Yu and Y. Yang, “Application of support vector regression machines to the processing of end effects of Hilbert-Huang transform,” *Mechanical Systems and Signal Processing,* vol. 21, no. 3, pp. 1197-1211, 2007. |
| [58] | Z. Peng and F. Chu, “Application of the wavelet transform in machine condition monitoring and fault diagnostics: a review with bibliography,” *Mechanical systems and signal processing,* vol. 18, no. 2, pp. 199-221, 2004. |
| [59] | P. Kankar, S. C. Sharma and S. Harsha, “Fault diagnosis of ball bearings using continuous wavelet transform,” *Applied Soft Computing,* vol. 11, no. 2, pp. 2300-2312, 2011. |
| [60] | I. a. M. G. Antoniadou, N. Dervilis, W. J. Staszewski and K. Worden, “On damage detection in wind turbine gearboxes using outlier analysis,” in *SPIE Smart Structures and Materials+ Nondestructive Evaluation and Health Monitoring*, International Society for Optics and Photonics, 2015, p. 83430N. |
| [61] | Q. Hu, Z. He, Z. Zhang and Y. Zi, “Fault diagnosis of rotating machinery based on improved wavelet package transform and SVMs ensemble,” *Mechanical Systems and Signal Processing,* vol. 21, no. 2, pp. 688-705, 2007. |
| [62] | F. Camci, K. Medjaher, N. Zerhouni and P. Nectoux, “Feature evaluation for effective bearing prognostics,” *Quality and reliability engineering international,* pp. 477-486, 2013. |
| [63] | M. Saimurugan, K. Ramachandran, V. Sugumaran and N. Sakthivel, “Multi component fault diagnosis of rotational mechanical system based on decision tree and support vector machine,” *Expert Systems with Applications,* pp. 3819-3826, 2011. |
| [64] | W. Caesarendra, A. Widodo and B.-S. Yang, “Application of relevance vector machine and logistic regression for machine degradation assessment,” *Mechanical Systems and Signal Processing,* vol. 24, no. 4, pp. 1161-1171, 2010. |
| [65] | T. W. Rauber, F. de Assis Boldt and F. M. Varejao, “Heterogeneous feature models and feature selection applied to bearing fault diagnosis,” *IEEE Transactions on Industrial Electronics,* vol. 62, no. 1, pp. 637-646, 2015. |
| [66] | B. Samanta, “Gear fault detection using artificial neural networks and support vector machines with genetic algorithms,” *Mechanical Systems and Signal Processing,* vol. 18, no. 3, pp. 625-644, 2004. |
| [67] | D. Chen and W. Wang, “Classification of wavelet map patterns using multi-layer neural networks for gear fault detection,” *Mechanical systems and signal processing,* vol. 16, no. 4, pp. 695-704, 2002. |
| [68] | B. Li, M.-Y. Chow, Y. Tipsuwan and J. C. Hung, “Neural-network-based motor rolling bearing fault diagnosis,” *IEEE Transactions on Industrial Electronics,* vol. 47, no. 5, pp. 1060-1069, 2000. |
| [69] | C. Murray, M. Asher, N. Lieven, M. Mulroy and C. a. M. P. Ng, “Wind turbine drivetrain health assessment using discrete wavelet transforms and an artificial neural network,” in *Renewable Power Generation Conference (RPG 2014), 3rd*, IET, 2014, pp. 1-5. |
| [70] | B. Samanta, K. Al-Balushi and S. Al-Araimi, “Artificial neural networks and support vector machines with genetic algorithm for bearing fault detection,” *Engineering Applications of Artificial Intelligence,* vol. 16, no. 7, pp. 657-665, 2003. |
| [71] | J. Rafiee, F. Arvani, A. Harifi and M. Sadeghi, “Intelligent condition monitoring of a gearbox using artificial neural network,” *Mechanical systems and signal processing,* vol. 21, no. 4, pp. 1746-1754, 2007. |
| [72] | M. Khazaee, H. Ahmadi, M. Omid, A. Banakar and A. Moosavian, “Feature-level fusion based on wavelet transform and artificial neural network for fault diagnosis of planetary gearbox using acoustic and vibration signals,” *Insight-Non-Destructive Testing and Condition Monitoring,* vol. 55, no. 6, pp. 323-330, 2013. |
| [73] | Z. Tian and M. J. Zuo, “Health condition prediction of gears using a recurrent neural network approach,” *IEEE Transactions on Reliability,* vol. 59, no. 4, pp. 700-705, 2010. |
| [74] | A. K. Mahamad, S. Saon and T. Hiyama, “Predicting remaining useful life of rotating machinery based artificial neural network,” *Computers & Mathematics with Applications,* vol. 60, no. 4, pp. 1078-1087, 2010. |
| [75] | N. Z. a. L. M. A. Gebraeel, “A neural network degradation model for computing and updating residual life distributions,” *IEEE Transactions on Automation Science and Engineering,* vol. 5, no. 1, pp. 154-163, 2008. |
| [76] | P. Konar and P. Chattopadhyay, “Bearing fault detection of induction motor using wavelet and Support Vector Machines (SVMs),” *Applied Soft Computing,* vol. 11, no. 6, pp. 4203-4211, 2011. |
| [77] | D. Lu and W. Qiao, “Adaptive feature extraction and SVM classification for real-time fault diagnosis of drivetrain gearboxes,” in *Energy Conversion Congress and Exposition (ECCE), 2013 IEEE*, 2013. |
| [78] | J. Kurek and S. Osowski, “Support vector machine for fault diagnosis of the broken rotor bars of squirrel-cage induction motor,” *Neural Computing and Applications,* vol. 19, no. 4, pp. 557-564, 2010. |
| [79] | F. a. F. G. a. T. C. a. V. P. Filippetti, “Recent developments of induction motor drives fault diagnosis using AI techniques,” *IEEE Transactions on Industrial Electronics,* vol. 47, no. 5, pp. 994-1004, 2000. |
| [80] | A. Kusiak and A. Verma, “A data-driven approach for monitoring blade pitch faults in wind turbines,” *IEEE Transactions on Sustainable Energy,* vol. 2, no. 1, pp. 87-96, 2011. |
| [81] | R.-t. f. d. f. g. t. g. s. u. e. l. machine, “Wong, Pak Kin ; Yang, Zhixin ; Vong, Chi Man ; Zhong, Jianhua,” *Neurocomputing,* vol. 128, pp. 249-257, 2014. |
| [82] | T.-H. Guo and J. Nurre, “Sensor failure detection and recovery by neural networks,” in *IJCNN-91-Seattle International Joint Conference on Neural Networks, 1991.*, Seattle, 1991. |
| [83] | J. C. da Silva, A. Saxena, E. Balaban and K. Goebel, “A knowledge-based system approach for sensor fault modeling, detection and mitigation,” *Expert Systems with Applications,* vol. 36, no. 12, pp. 10977--10989, 2012. |
| [84] | C. Sobie, C. Freitas and M. Nicolai, “Simulation-driven machine learning: Bearing fault classification,” *Mechanical Systems and Signal Processing,* no. 99, pp. 403-419, 2018. |
| [85] | K. Hendrickx, W. Meert, B. Cornelis, K. Janssens, K. Gryllias and J. Davis, “A fleet-wide approach for condition monitoring of similar machines using time-series clustering,” in *Int. Conf. on Condition Monitoring of Machinery in Non-stationary operations (CMMNO)*, Santander, Spain, 2018. |
| [86] | B. H. McCormick, T. DeFanti and M. Brown, “Visualization in Scientific Computing,” *Computer Graphics,* vol. 21, no. 6, 1987. |
| [87] | J. Larkin and H. A. Simon, “Why a Diagram is (Sometimes) Worth ten thousand words,” *Cognitive Science,* vol. 11, pp. 65-99, 1987. |
| [88] | H. Schumann and W. Muller, Visualisierung – Grundlagen und allgemeine Methoden, Berlin: Springer, 2000. |
| [89] | W. Aigner, S. Miksch, H. Schumann and C. Tominski, Visualization of time-oriented data, London: Springer, 2011. |
| [90] | J. Bertin, Semilogy: Diagrams, Networks, Maps, Madison, WI, USA: University of Wisconsin Press, 1983. |
| [91] | J. Mackinlay, “Automating the Design of Graphical Presentations of Relational Information,” *ACM Transactions on Graphics,* vol. 5, no. 2, pp. 110-141, 1986. |
| [92] | J. Mennis, D. Peuquet and L. Qian, “A Conceptual Framework for Incorporating Cognitive,” *International Journal of Geographical,* vol. 25, no. 6, pp. 501-520, 2000. |
| [93] | S. Havre, E. Hetzler, P. Whitney and L. Nowell, “ThemeRiver: Visualizing Thematic,” *IEEE Transactions on Visualization and Computer,* vol. 8, no. 1, pp. 9-20, 2002. |
| [94] | J. Wise, J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur and V. Crow, “Visualizing the non-visual: Spatial analysis and interaction with information from text documents,” *Proceedings of the Symposium on Information Visualization,* pp. 51-58, 1995. |
| [95] | C. Chen, Information visualisation and virtual environments, London: Springer-Verlag, 1999. |
| [96] | M. Dodge, “Web visualization,” October 2001. [Online]. Available: http://www.geog.ucl.ac.uk/casa/martin/geography\_of\_cyberspace.html. |
| [97] | M. Ward, G. Grinstein and D. Keim , Interactive data visualization: foundations, techniques and applications, Natick, MA, USA: A K Peters Ltd, 2010. |
| [98] | J. Yi, Y. ah Kang, J. Stasko and J. Jacko, “Toward a Deeper Understanding of the Role of Interaction in Information Visualization.,” *IEEE Transactions on Visualization and Computer Graphics,* vol. 13, no. 6, pp. 1224-1231, 2007. |
| [99] | “SynerScope,” SynerScope B.V., [Online]. Available: www.synerscope.com. |
| [100] | R. Harris, Information Graphics: A Comprehensive Illustrated Reference, New York, NY, USA: Oxford University, 1999. |
| [101] | E. Tufte, Beautiful Evidence, Cheshire, CT: Graphics Press, 2006. |
| [102] | D. Brodbeck and L. Girardin, “Interactive Poster: Trend Analysis in Large Timeseries of High-Throughput Screening Data Using a Distortion-Oriented Lens with Semantic Zooming,” in *In Poster Compendium of IEEE Symposium on Information Visualization (InfoVis)*, Los Alamitos, CA, USA, IEEE Computer Society, 2003, pp. 74-75. |
| [103] | S. Card, B. Pendelton, J. Heer and J. Bodnar, “Time Tree: Exploring Time Changing Hierarchies,” in *In Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)*, Los Alamitos, CA, USA, IEEE Computer Society, 2006, pp. 3-10. |
| [104] | R. Erbacher, K. Walker and D. Frincke, “Intrusion andMisuse Detection in Large-scale systems,” *IEEE Computer Graphics and Applications,* vol. 22, no. 1, pp. 38-48, 2002. |
| [105] | D. Mintz, T. Fitz-Simons and M. Wayland, “Tracking Air Quality Trends with SAS/GRAPH,” in *In Proceedings of the 22nd Annual SAS User Group International Conference (SUGI97)*, Cary, NC, USA, 1997. |
| [106] | K. Beard , E. Heather and N. Pettigrew, “A Framework for Visualization and Exploration of events,” *Information Visualization,* vol. 7, pp. 133-151, 2008. |
| [107] | P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández and E. Vázquez, “Anomaly-based network intrusion detection: Techniques, systems and challenges,” *Comput. Secur.,* vol. 28, pp. 18-28, 2009. |
| [108] | Y. Zhang, N. Meratnia and P. Havinga, “Outlier detection techniques for wireless sensor networks: A survey,” *Commun. Surv. Tutor. IEEE,* vol. 12, pp. 159-170, 2010. |
| [109] | M. A. Hayes and M. A. Capretz, “Contextual anomaly detection in big sensor data,” *IEEE Int. Congr. On Big Data,* pp. 64-71, 2014. |
| [110] | Y. Kou, C. T. Lu, S. Sirwongwattana and Y. P. Hua, “Survey of fraud detection techniques,” *IEEE international conference on networking, sensing and control,* pp. 749-754, 2004. |
| [111] | T. Rabenoro, J. Lacaille, M. Cottrell and F. Ross, “A Methodology for the Diagnostic of Aircraft Engine Based on Indicators Aggregation,” in *Advances in Data Mining. Applications and Theoretical Aspects*, Springer International Publishing, 2014, pp. 144-158. |
| [112] | S. Verron, T. Tiplica and A. Kobi, “Procedure based on mutual information and bayesian networks for the fault diagnosis of industrial systems,” *American Control Conference,* pp. 07-420, 2007. |
| [113] | V. Chandola, A. Banerjee and V. Kumar, “Anomaly detection: A survey,” *ACM Comput. Surv. CSUR,* vol. 41, p. 15, 2009. |
| [114] | M. Amer and M. Goldstein, “Nearest-neighbor and clustering based anomaly detection algorithms for rapidminer,” *Proc. of the 3rd RapidMiner Community Meeting and Conference (RCOMM),* pp. 1-12, 2012. |
| [115] | L. Friedland, A. Gentzel and D. Jensen, “Classifier-Adjusted Density Estimation for Anomaly Detection and One-Class Classification,” *SDM,* pp. 578-586, 2014. |
| [116] | M. Gupta, J. Gao, C. C. Aggarwal and J. Han, “Outlier Detection for Temporal Data: A Survey,” *IEEE Trans. Knowl. Data Eng.,* vol. 26, no. 9, pp. 2250-2267, 2014. |
| [117] | D. J. Hill, B. S. Minsker and E. Amir, “Real-time Bayesian anomaly detection in streaming environmental data,” *Water Resour. Res.,* vol. 45, no. 4, p. W00D28, Apr. 2009. |
| [118] | M. Breunig, H. -P. Kriegel, R. T. Ng and J. Sander, “LOF: Identifying Density-Based Local Outliers,” *PROCEEDINGS OF THE 2000 ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA,* pp. 93-104, 2000. |
| [119] | J. Lin, E. Keogh, A. Fu and H. V. Herle, “Approximations to magic: finding unusual medical time series,” *IEEE Symposium on Computer-Based Medical Systems (CBMS’05),* pp. 329-334, 2005. |
| [120] | W. Dickinson, D. Leon and A. Fodgurski, “Finding failures by cluster analysis of execution profiles,” pp. 339-348, 2001. |
| [121] | A. Mirgorodskiy, N. Maruyama and B. Miller, ““Problem Diagnosis in Large-Scale Computing Environments,” p. 11, 2006. |
| [122] | W. Xu, L. Huang, A. Fox, D. Patterson and M. I. Jordan, “Detecting large-scale system problems by mining console logs,” *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles,* pp. 117-132, 2009. |
| [123] | D. Cotroneo, R. Pietrantuono, L. Mariani and F. Pastore, “Investigation of failure causes in workload-driven reliability testing,” *Fourth international workshop on Software quality assurance: in conjunction with the 6th ESEC/FSE joint meeting,* pp. 78-85, 2007. |
| [124] | S. Orlando and S. Russo, “Java Virtual Machine Monitoring for Dependability Benchmarking,” pp. 443-440, 2006. |
| [125] | J. Tan, X. Pan, S. Kavulya, R. Gandhi and P. Narasimhan, “SALSA: Analyzing Logs as StAte Machines,” *WASL,* vol. 8, p. 6, 2008. |
| [126] | N. Palatin, A. Leizarowitz, A. Schuster and R. Wolff, “Mining for misconfigured machines in grid systems,” *Data Min. Tech. Grid Comput. Environ.,* p. 71, 2008. |
| [127] | M. Chen, A. X. Zheng, J. Lloyd, M. I. Jordan and E. Brewer, “Failure diagnosis using decision trees,” *Proceedings. International Conference on Autonomic Computing,* pp. 36-43, 2004. |
| [128] | W. B. Cavnar, J. M. Trenkle and others, “N-gram-based text categorization,” *Ann Arbor MI,* vol. 48113, no. 2, pp. 161-175, 1994. |
| [129] | K. Spärck Jones, “A statistical interpretation of term specificity and its application in retrieval,” *J. Doc.,* vol. 60, no. 5, pp. 493-502, Oct. 2004. |
| [130] | Y. Guan, A. -A. Ghorbani and N. Belacel, “Y-means: A clustering method for intrusion detection,” 2003. |
| [131] | M. Ester, H. -P. Kriegel, J. Sander, X. Xu and others, “A density-based algorithm for discovering clusters in large spatial databases with noise.,” *Kdd,* vol. 96, pp. 226-231, 1996. |
| [132] | A. Jain and D. Zongker, “Feature selection: Evaluation, application, and small sample performance,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* pp. 153-158, 1997. |
| [133] | P. J. Dempsey, “A comparison of vibration and oil debris gear damage detection methods applied to pitting damage.,” 200. |
| [134] | Y. Shao and K. Nezu, “Prognosis of remaining bearing life using neural networks,” *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering,* vol. 214, no. 3, pp. 217-230, 2000. |
| [135] | J. Wang, Numerical and experimental analysis of spur gears in mesh, 2003. |
| [136] | S. Sirichai, Torsional properties of spur gears in mesh using nonlinear finite element analysis., 1999. |
| [137] | E. Tufte, The Visual Display of Quantitative Information, Cheshire, CT: Graphics Press, 1983. |
| [138] | E. Bier, M. Stone, K. Pier and W. Buxton, “Tool glass and magic lenses: The see-through,” *Computer,* pp. 73-80, 1993. |

1. RAS = Reliability, Availability and Serviceability [↑](#footnote-ref-1)
2. MTBF = Mean Time Between Failure [↑](#footnote-ref-2)
3. The “light-lease” concept entails selling or leasing light to cinema exhibitors instead of selling digital cinema projector lamps [↑](#footnote-ref-3)
4. <https://www.barco.com/en/product/nms> [↑](#footnote-ref-4)
5. Elasticsearch, Logstash, Kibana [↑](#footnote-ref-5)