

# DeepFMEA – A Scalable Framework Harmonizing Process Expertise and Data-Driven PHM

Christoph Netsch<sup>1</sup>, Till Schöpe<sup>1</sup>, Benedikt Schindele<sup>1</sup>, and Joyam Jayakumar<sup>1</sup>

<sup>1</sup>*Alpamayo Intelligent Quality Solutions, Sarnen, 6060, Switzerland*  
{netsch/schoepe/schindele/jayakumar}@alpamayo.ch

## ABSTRACT

In an industrial setting, data is often limited in quantity, and its quality can be inconsistent. Yet, the potential of Machine Learning (ML) based prognostics and health monitoring (PHM) tools to enable risk-optimized operations and maintenance relies heavily on these two aspects. In practice, to bridge this gap, successful developments heavily rely on the introduction of process/domain expertise as a prior, enabling sufficiently accurate predictions, while enhancing their interpretability.

A key challenge in development involves translating the experience and process knowledge of maintenance personnel, development, and service engineers into a data structure. This structure must not only capture the vast diversity and variability of the expertise but also render this knowledge accessible for various data-driven algorithms.

These challenges result in data models that are heavily tailored towards a specific application and the failure modes the development team aims to detect and predict via ML. The resulting lack of a standardized modelling approach limits developments' extensibility to new failure modes, their transferability to new applications, and it inhibits the utilization of standard data management and MLOps tools, further increasing the burden on the development team. In effect, the resulting high development and industrialization costs limit the economic utility of data-driven PHM tools to use cases with an exceptionally high economic risk.

DeepFMEA, draws inspiration from the Failure Mode and Effects Analysis (FMEA) in its structured approach to the analysis of any technical system and the resulting standardized data model, while considering aspects that are crucial to capturing process and maintenance expertise in a way that is both intuitive to domain experts and the resulting information can be introduced as priors to Machine Learning algorithms. In effect, our proposed framework promises a consistent use of best practices in data-driven modeling for PHM use cases while enhancing their scalability, interpretability, and cost-effectiveness.

## 1. INTRODUCTION

The widespread adoption of data-driven technologies has opened new horizons in various industrial applications, among which prognostics and health monitoring (PHM) stands out as a critical area of focus. PHM refers to the use of advanced analytical tools and techniques for online monitoring of equipment, diagnostics of specific failure modes, and prognostics regarding the future performance or failure of machinery and equipment. By leveraging Machine Learning (ML) and other data-driven methodologies, PHM aims to enable the adoption of condition-based and predictive maintenance techniques. (Nunes et al., 2023) This promises to optimize maintenance schedules and minimize the level of risk resulting from the consequences of failures, such as operational downtime or reduced energy efficiency, thereby enhancing overall system reliability and efficiency.

Original equipment manufacturers (OEMs), system integrators, maintenance and repair organizations (MROs) and operators of large fleets are each well-positioned to develop and deploy PHM tools for a specific class of equipment. To varying degrees, they possess:

- access to in-operation data representative of the operational context, under which the equipment is operated, and to maintenance and service data,
- an understanding of the equipment's design and its dominant failure modes and the experience, allowing for reasonable prior assumptions regarding consequences and failure rates,
- an understanding of the maintenance policies in place, and the requirements that determine how a PHM tool can be utilized to enable condition-based and predictive maintenance techniques.

OEMs and system integrators in particular are in a unique position to integrate these solutions natively into equipment at the point of production, promising close alignment

---

Till Schöpe, Christoph Netsch, Benedikt Schindele, Joyam Jayakumar. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided

between the design of the machinery and the PHM tool, including the use of the full range of available signals from intrinsic sensors and control signals in a system and the ability to embed sensors that cover potential “gray spots” of a system. In promising improved asset reliability and longevity as well as increased maintenance efficiency, they intend to create differentiating characteristics for their product, their spare parts offering, competitive advantages for their aftersales services, or entirely new business models around their core product (Potthoff et al., 2023).

Reliability-centered maintenance (RCM) (Basson, 2019) and Total Productive Maintenance (TPM) (Wireman, 2004), among others provide robust theories on the circumstances, under which adopting a condition-based or predictive maintenance policy is both practical and cost-effective. However, given the imperfect nature of their predictions, the overall risk reduction potential of a PHM tool cannot be assessed without understanding how its deployment would impact the tasks realized by maintenance practitioners. This can be characterized by three key factors:

- the **accuracy** of that information, with respect to the rates of true detections, false detections, and missed detections, influencing savings related to avoided failure consequences and the additional costs of unnecessary diagnostic tasks.
- the **prediction horizon** characterized by the time between the detection of a potential failure and the observance of the corresponding functional failure, which corresponds to the P-F-interval for many failure patterns and determines the practicality of a proactive task.
- the **prescriptive value** of the information provided to the maintenance practitioner – a result of both the choice of algorithm and the interpretability of its output - influencing how directed the resulting task is.

Based on requirements stemming from a specific use case, and the skillsets and convictions of a particular team tasked with developing such systems, teams tend to make use of data-driven modeling techniques (classical statistical models and machine learning) or mechanistic modeling techniques to varying degrees. Mechanistic models are trusted for their explicit representation of domain knowledge and the interpretability of their outputs. However, with increasing complexity of a system and the uncertainty related to stochastic phenomena and incomplete knowledge of a system’s operational context and environment, the gap between model and reality renders purely mechanistic models useless for many use cases (Eker et al., 2016; Hagemeyer et al., 2022)

Data-driven models, on the other hand, are able to capture complex relationships within data, given an incomplete representation of a specific asset’s history and operational

context, can frequently be refitted to new data, and, in theory, require no prior assumptions (Liao et al., 2016). However, the outputs created by purely data-driven models rarely provide the prescriptive value that condition-based and predictive maintenance demands for and enjoy less trust among maintenance practitioners, due to their lack of interpretability (Vollert et al., 2021). Additionally, in the context of PHM, the utility of purely data-driven modeling is severely limited, due to the variability and quality of the data upon which these systems rely (Luo et al., 2020, Nunes et al., 2023) and their limited availability - particularly in industrial applications, where data sharing collaborations between different organizations remain an exception (Trauth et al., 2020).

In practice, and with the exception of some examples of purely academically motivated research and proof-of-concept (PoC) implementations, the underlying models of PHM tools are often hybrid in nature (Luo et al., 2020). On one hand, hybrid modelling benefits from the adaptability and ability of data-driven models to learn from incomplete information. On the other hand, it leverages domain knowledge to enhance output interpretability and introduce priors into the model that compensate for a data-driven model’s ability to generalize to operating contexts and failures, that are underrepresented in the available data. It is not surprising that the PHM community in particular has proposed noteworthy contributions to the field of hybrid machine learning, for instance the use of graph neural networks to learn representations of the semantic relationships of signals inherent to complex technical systems and to introduce known relationships as priors to a neural network (Battaglia et al., 2018, Zhao et al., 2020) or approaches designed to make connect available, yet incomplete or inaccurate physics-based models with ML-based models (Gassner et al., 2014, Arias Chao et al., 2022)

Our contribution does not challenge current fundamental contributions to hybrid modeling, nor does it extend the field’s state-of-the-art. Rather, by proposing DeepFMEA as a standardized framework for the development and deployment of data-driven PHM tools our contribution recognizes the importance of these contributions, while strengthening the basis for their industrialization. To do so, in *section 2* we first derive requirements for a potential framework from current challenges. In *section 3*, we propose (3.1) a standardized data model allowing for a structured representation of domain knowledge in PHM use cases, (3.2) a non-comprehensive overview of methods to harmonize this information in hybrid data-driven modeling approaches, and (3.3) an overview of methods to enrich model outputs with prescriptive information, to enhance their actionability. In *section 4* we demonstrate an example of DeepFMEA usage in a practical use case based on monitoring and diagnostics for a hydraulic system (Helwig, 2015). Finally, *section 5* summarizes our findings from implementations of the framework in real-world PHM projects and discusses its current limitations and extension potentials.

## 2. CHALLENGES IN THE INDUSTRIALIZATION OF DATA-DRIVEN PROGNOSTICS AND HEALTH MONITORING TOOLS

In practice, the main effort of developing PHM solutions is not spend with model development. In order to deploy and operate data-driven PHM tools in a production environment, substantial development effort is introduced by the data and software engineering required to provide the following functions:

- **Data management** encompasses IoT-connectivity, data pipelines, centralized or decentralized databases, and data processing services (data health checks, preprocessing, inference, and postprocessing) that enable the tool's online operation.
- **Model monitoring & updates** implements feedback loops that automate the continuous evaluation of the tool's underlying model in production by relating its outputs to any available maintenance and operations data from a given asset. Furthermore, this encompasses the model training service, model registry, and decision logic required to adapt, update, and replace models in a production environment. Despite these capabilities falling into the category of MLOps (Kreuzberger et al., 2023) arguably, in PHM, they are not only required to operate ML-models, but data-driven models in general. This is due to the fact that datasets, PHM tools are initially developed with, rarely reflect the full range of operating contexts and failures, that a fleet of assets may experience in operation (Zio, 2022)
- **The process integration layer** includes reporting services and interfaces to enterprise software (i.e. ERP, CMMS) needed to make model outputs accessible to asset managers, field engineers, and maintenance professionals, i.e. as decision-support systems or automations. An additional function realized in the application layer of a PHM solution is to capture the maintenance and operations data required to realize the monitoring function.

The heavy reliance on domain knowledge, described in *section 1*, combined with the fact that PHM tools frequently rely on the ingestion and fusion of control level (PLC), process management level (SCADA), and/or management level (MES) data (Mantravi et al., 2022) commonly results in highly customized data models and data pipelines. While most requirements of the aforementioned functions can be abstracted to a level, where they are agnostic to a specific PHM use case, this high degree of customization introduces additional requirements for the data management, MLOps, and process integration layers that interface with them.

In effect, this inhibits the use of general-purpose software modules and increases the burden on the development team. Particularly the implementation of MLOps functions

demands a highly specialized skillset (Nahar, 2022) which is rarely represented in PHM development teams.

Considering the prevalence of hybrid modeling approaches described in *section 1* and the challenges to their industrialization introduced in section 2, a framework for the development of data-driven PHM tools requires:

- a consistent representation of domain knowledge commonly relevant to the PHM use case, such as:
  - semantic relationships between sensors and the elements of a complex technical system they are localized to,
  - prior knowledge that facilitates the diagnostic and prognostic functions of a PHM tool with respect to specific failure modes, i.e. virtual sensors, detection rules, and degradation variables,
  - prescriptive information, i.e. recommended diagnostic, proactive, or reactive maintenance tasks, given process anomalies or a specific failure mode is detected,
- a data structure and systematic approach that connects in-operation data, maintenance data, and domain expertise via hybrid modeling approaches, yet maintaining the flexibility to integrate problem-specific approaches or advances in the state-of-the-art,
- a common approach towards how data-driven monitoring tools for the purposes of monitoring, diagnostics, and prognostics are managed in production, to facilitate the efficient use of general-purpose technology,
- a quantitative assessment of monitoring, diagnostics, and prognostics tools that quantifies their impact in terms of risk-reduction, considering the imperfect nature and the inherent uncertainty of data-driven model predictions.

## 3. CONCEPT

*DeepFMEA*, our proposed framework, does not draw its name being an implementation the Failure Modes and Effects Analysis (FMEA) (MIL-STD-1629A, 1980, Rausand et al., 2003) nor is it a data-driven extension to the FMEA process, as proposed by (Ervural & Ayaz, 2023). We draw inspiration from:

- the strong intersection between the data required by the FMEA and the information commonly used as domain expertise in the modeling process,

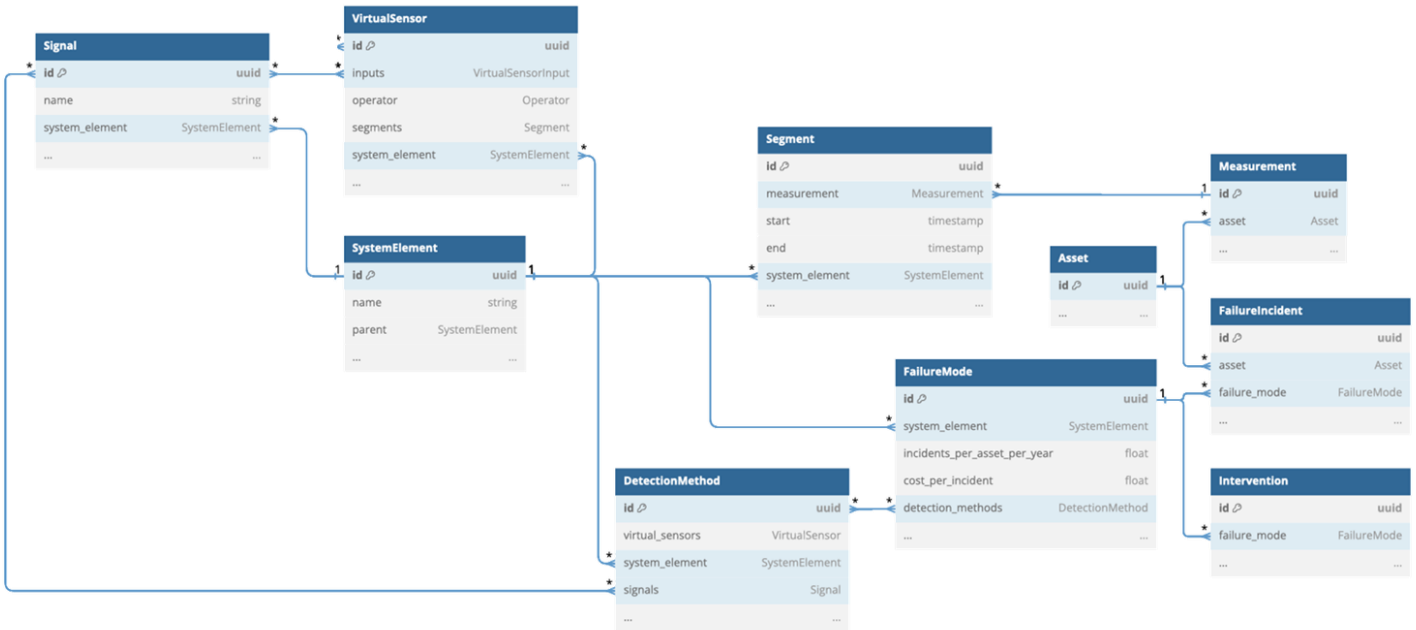


Figure 1: Simplified UML Diagram describing the data models used within the DeepFMEA Framework.

- the systematic process of capturing this data within an FMEA,
- the structured relational data model suggested by the analysis, which has been implemented in database schemas by numerous vendors of specialized FMEA-software.

This section briefly outlines the abstract data model implemented by *DeepFMEA* (3.1), before presenting a non-comprehensive collection of approaches to systematically utilize this information in the modeling process (3.2), and enriching model outputs with prescriptive information and assessing their impact on risk (3.3).

### 3.1. Data models

A *System Element* reflects the part of a system at which a failed state (the inability of a system to fulfill its intended function) can be localized at. Deconstructing a complex system into system elements results in a hierarchical data structure, frequently resembling the subsystems, assemblies, components, or parts of its physical twin. The root system element corresponds to the system modelled in the PHM project.

An *Asset* corresponds to a physical entity of the root system element.

A *Signal* declares any class of in-operation data captured in a technical system, either from intrinsic sensors, control signals, or external sensors retrofitted to the system. A *Signal* references one or multiple *System Elements*.

A *Measurement* contains the time-series associated referencing a *Signal* of an *Asset* in a given time interval. Whenever a system’s process is cyclic in nature, it is a good practice to define *Measurements* along the boundaries of cycles, since this already introduces strong normalization to the time-series simplifying their subsequent processing.

A *Segment* references the method and its parameters required to retrieve a given recurring pattern, i.e. corresponding to a specific step in a process or a procedure in the operation of a machine. It is model-agnostic. The underlying segmentation model could be a simple rule, a motif detection algorithm or even a neural network.

A *Virtual Sensor* defines the computation of a property of one or multiple *Measurements*, which is informed by domain knowledge. *Virtual Sensors* are commonly used in PHM projects due to their enhanced interpretability, expressiveness and better generalization properties compared to raw signals. An example of a *Virtual Sensor* in a shaft component is the “1X Frequency”. Elevated levels are commonly (but not exclusively) associated with its misalignment. In analogy to a *Signal*, it references one or multiple *System Elements*. The virtual *Measurements* obtained can be scalar, vectorial or tensorial. The computation of a *Virtual Sensor* is stored as a graph of atomic *Operations*, that reference *Signals*, *Segments*, or other *Virtual Sensors*. Section 3.2 introduces multiple concepts of how *Virtual Sensors* can be used in monitoring, diagnostics, and prognostics *Detection Methods*.

A *Failure Mode* defines one form in which a *System Element* can fail. In order to enable an assessment of risk and the risk-

reduction potential corresponding to the deployment of a *Detection Method*, a *Failure Mode* includes properties that quantify the (economic) consequences of a *Failure Incident*. For the same purpose, due to the fact that *Failure Incidents* are rare in many PHM use cases, in practice, prior assumptions regarding the failure rate may have to be introduced as additional properties. Reasonable assumptions should be set in agreement between multiple domain experts.

An *Intervention* stores a prescriptive instruction directed to an asset's operator or maintainer, given the occurrence of a specific *Failure Mode*. An *Intervention* can be a diagnostic, reactive, or proactive task.

A *Failure Incident* stores a detected or observed occurrence of a *Failure Mode* for a given *Asset*. In order to monitor the performance of a given *Detection Method*, its properties allow determining whether the *Failure Incident* was detected prior to its occurrence, remained undetected or was a false alarm.

A *Detection Method* references any monitoring, diagnostic, or prognostic model developed and deployed as part of the PHM tool. It carries references to both *Signals* and *Virtual Sensors*, that it relies on as inputs, and to the *System Elements* (in the case of monitoring) or *Failure Modes* (in the case of diagnostics and prognostics).

### 3.2. Systematic Use of Domain Knowledge in Detection Methods

Below, we present a selection of methods that illustrate how the data mentioned earlier can be systematically leveraged to develop data-driven PHM tools. Instead of categorizing these methods by specific algorithm classes, we organize our discussion around the typical progression of objectives in PHM projects. Projects often start with Proof of Concepts (PoCs) that focus primarily on monitoring the operation of assets by detecting deviations from normal processes. They then gradually extend the tool's capabilities to diagnose high-risk failure modes and, eventually, to forecast potential incidents. By examining the data requirements for each of these objectives, *DeepFMEA* ensures that a PHM tool can be expanded without significant alterations to its data and software architecture and the high-quality data, that is particularly important for diagnostics and prognostics, is collected from the onset of the project.

#### 3.2.1 Monitoring

Monitoring targets the detection of anomalous behavior in a process without pinpointing specific failure modes. employs anomaly detection algorithms that leverage readily available data from normal operations. Due to the complexity and high dimensionality of data from industrial equipment, direct application of data-driven models often leads to poor generalization. Moreover, it inhibits the ability to localize anomalies within a large system. Therefore, it is common

practice to filter data and distill only the most pertinent information.

The data model defines the relationships between the *Detection Method*, the *System Element(s)* within its scope, and the associated *Signals*. It permits a preselection of data relevant to the monitoring objective while eliminating signals that do not enhance the monitoring tool's performance because they lack informational value about the focused *System Elements*.

Moreover, *Virtual Sensors*, associated with the *System Elements* under observation, can be input to anomaly detection models as Health Indicators. These indicators extract crucial information from the dense, noisy, and often overly detailed sensor data collected from real-world assets, ensuring the monitoring process is both efficient and effective.

#### 3.2.2 Diagnostics

Unlike monitoring, diagnostic tools offer operators and maintainers precise information that identifies a failure mode or several probable ones. Building on monitoring concepts, the data model's embedded domain expertise can be further utilized for diagnostic challenges.

Diagnostics typically takes the form of a classification problem, where *Failure Modes* defined in the data model represent the classes. Implementing this model in a database together with the data pipelines, that make the data accessible from its sources, results in a scalable data management system, aligning *Failure Incidents* with their respective *Failure Mode*. The continuous collection of failure data not only provides the data required to fit the diagnostic *Detection Method's* model or evaluate its performance metrics; it also creates the basis for automated monitoring, adaption, and updating of any data-driven models in a production environment.

Diagnostics use cases often face a notable lack of Failure Incidents, particularly for high-risk Failure Modes. To address this, PHM projects might rely on strong assumptions, like establishing explicit or inferred thresholds on *Virtual Sensor* values for a *Failure Mode's* occurrence, based on expert consensus.

Additionally, section 4 illustrates a simplified example showing how data on failure rates, consequences, and interventions can evaluate a *Detection Method's* risk reduction impact upon deployment.

#### 3.2.3 Prognostics

Prognostics aims to forecast specific Failure Modes by estimating the remaining useful life (RUL) of a *System Element*. Although data-driven models for prognostics have been a focal point in PHM research, their practical impact on industrial PHM tools remains minimal. The prominence of research in this area within parts of the PHM community

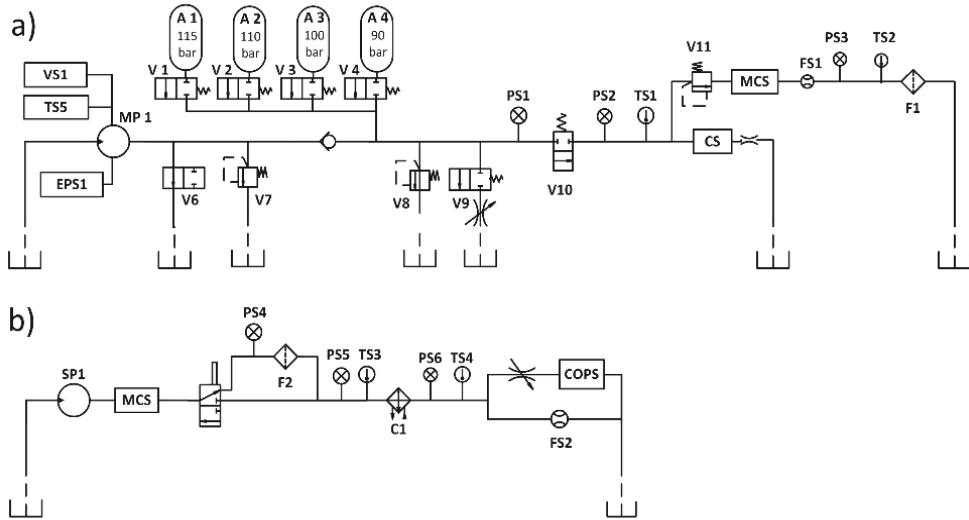


Figure 2. Hydraulic system testbench used to obtain training data for (Helwig, 2018).

might give the impression of their significant role, but in practice, the utility of RUL predictions is often limited. Despite their allure, they may be of limited value for many *Failure Modes* where monitoring or diagnostic tools can identify potential failures well in advance of the functional failure, allowing for their proactive management. Furthermore, for *Failure Modes* governed by stochastic events, is a futile endeavor.

Creating effective data-driven prognostics models requires extensive datasets, including both numerous *Failure Instances* and comprehensive in-operation data on the degradation trajectory. Such complete datasets are exceptionally rare in practice. However, a data management system implementing the *DeepFMEA* data model has the technical capability to compile datasets fulfilling these stringent criteria. When a *Failure Mode* is thoroughly understood, domain experts can identify *Virtual Sensors* as degradation variables, feeding into models that estimate RUL.

### 3.3. Enhancing the Prescriptive Value of PHM Outputs

Data-driven models often face criticism for their lack of interpretability. However, the rich contextualization the *DeepFMEA* offers for a *Detection Method* can significantly enhance the model's outputs. By adding context data and prescriptive information before presenting results to operators and maintainers, the trustworthiness of a PHM tool is improved, making it a more valuable decision-support system.

When a deviation from the healthy process is detected, a monitoring tool can use the available information on the *System Element(s)* referenced by the *Detection Method* to provide suggestions for its origins and reduce the amount of time for its localization.

The presentation of context data can mirror the hierarchical structure of the physical system, allowing operators and maintainers to intuitively trace anomalies from a system-wide perspective down to specific components. This structured and visualized approach aligns with the natural process of investigating anomalies.

Furthermore, diagnostic, proactive, and reactive *Interventions* tied to particular *Failure Modes* can be directly communicated to the operator or maintainer. Alternatively, they can be seamlessly integrated into existing processes, automatically initiating the appropriate workflows. This integration facilitates a more effective and timely response to potential issues.

### 4. REFERENCE USE CASE: HYDRAULIC SYSTEM DATASET

To demonstrate the use of the proposed *DeepFMEA* framework, we show the integration of Helwig's hydraulic system dataset (Helwig, 2018). This dataset was obtained by a hydraulic test rig where different degradation and multiple failure modes were simulated. The experimental setup, consisting of a working circuit including a variable load and a cooling and filtration circuit can be found in Figure 1. The rig is equipped with pressure, temperature, vibration and volume flow sensors at different components. Virtual Sensors, Features and analytics methods to detect the simulated *Failure Modes* are proposed (Helwig, 2015).

We translate the *Condition Monitoring* approach described in Helwig’s paper into our *DeepFMEA* framework by describing and linking the system-components with sensors using the *System Element* and *Signal* data models. We then add the expert knowledge found in the form of *Virtual Sensors* and known *Failure Modes* as well as feature engineering performed in the paper. It should be noted that both feature engineering as well as the virtual sensors can be united into the same data-model *Virtual Sensor* within our framework.

We start by describing the physical machine as a tree-structure of *System Elements*. We then continue to link all available *Signals* to their respective *System Element*. This hierarchical linking for example allows for the aggregation of *Signals* while also considering child components. Even though EPS1 and VS1 are not referencing the exact same *System Element*, they can all be aggregated to the parent component “Pump”.

After describing the hydraulic rig within the proposed framework, we continue by adding the data models describing the condition-based monitoring solutions. The first step is to define the different *Failure Modes* to be detected. In this case, they are defined by the *Failure Modes* the experimental setup is able to simulate: Cooling power decrease, Switching characteristic degradation of the valve, internal leakage of the pump and gas leakage of one of the accumulators. As seen in Table 1, referencing the corresponding system element is straightforward for each of the *Failure Modes*.

In order to improve the detection performance of the *Detection Methods*, Helwig 2015 preprocesses the raw sensor data by defining data *Segments* as well as *Virtual Sensors*. Both are part of the expert knowledge enriching PHM solutions. The segments, or “intervals”, are defined by the

variable load on the test rig, the virtual sensors consist of physical relevant knowledge about cooler (cooling efficiency, cooling power), using in depth engineering knowledge like the heat transport equation of oil.

Table 5 shows how both feature engineering and virtual sensors can be united in the same data model *Virtual Sensor* with cooling efficiency and two median features as an example. As described in the original paper (Helwig, 2015), the cooling efficiency virtual sensor is the ratio between the oil temperature difference before and after the cooler and the difference of the oil temperature before the cooler and the ambient temperature. We split the computation into multiple atomic mathematical operations, i.e. the calculation of both temperature differences followed by the building the ratio. We apply the virtual sensor on segment 1, as it depicts the whole working cycle.

Incorporating feature engineering in the data model works analogously: A mathematical atomic operation as well as an input and a segment are defined. The input can either be another *Virtual Sensor* or a *Signal*.

With the equipment, signals and expert knowledge described in a structured, contextualized manner, every information needed to create PHM solutions for the hydraulic test rig from a technical point of view is now present in the *DeepFMEA* framework. Figure 4 shows an exemplary visualization of an unsupervised machine learning model trained to perform *Diagnostics* for the defined *Failure Modes*, where the labels of the dataset, where incorporated as *Failure Incidents* and *Interventions*.

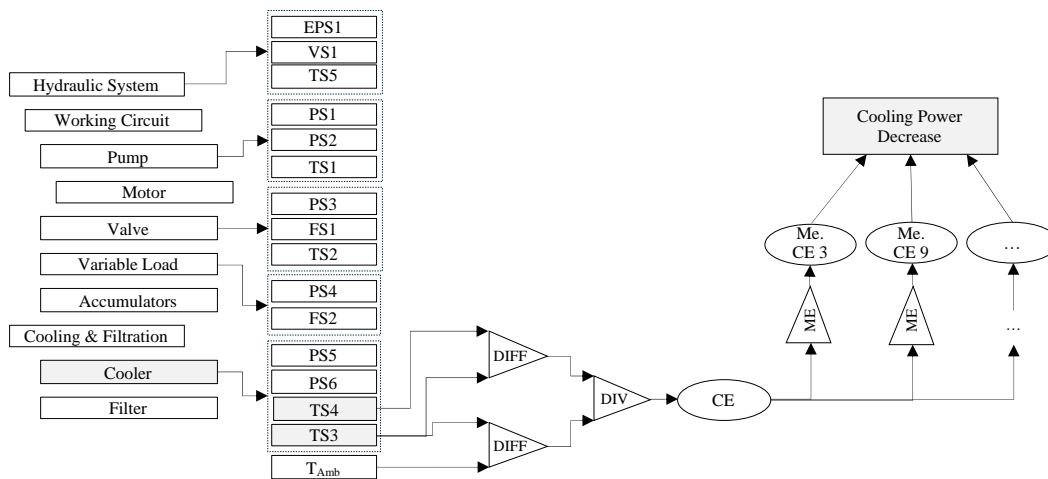


Figure 3. Visual description of the hydraulic systems including links used within the DeepFMEA data structure with the example of “Cooling Power Decreased” as Failure Mode.

Table 1. Simplified System Element data-model.

Name	Parent (FK)
Hydraulic-System	NULL
Working-Circuit	Hydraulic-System
Pump	Working-Circuit
Motor	Pump
Valve	Working-Circuit
Variable Load	Working-Circuit
Accumulators	Cooling & Filtration
Cooling & Filtration	Working-Circuit
Cooler	Cooling & Filtration

Table 2: Simplified signal data-model.

Name	System Element (FK)	Sampling-Rate
EPS1	Motor	100 Hz
VS1	Pump	1 Hz
PS1	Valve	100 Hz
...		
TS3	Cooler	1 Hz
TS4	Cooler	1 Hz

Table 3. Simplified Segment data-model.

Name	Start	End
INT1	0.00s	60.00s
...		
INT13	50.01	60.00s

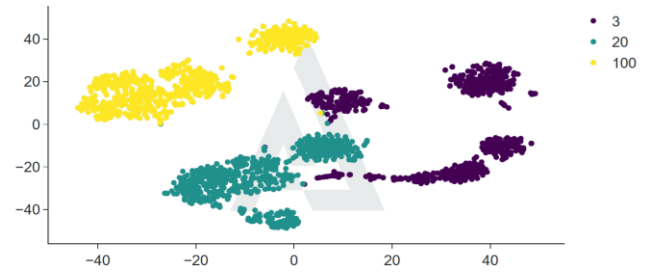
Table 4. Simplified Failure Mode data model.

Name	System Element (FK)
Cooling Power Decrease	Cooler
Internal Leakage	Pump
Gas Leakage	Accumulators
Switching char. degradation	Valve

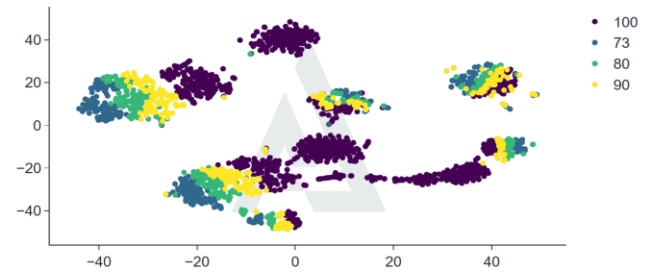
Table 5. Simplified Virtual Sensor data-model.

Name	Inputs	Operator
$\Delta T_{Cool}$	(T3, INT1) (T4, INT1)	DIFF
$\Delta T_{Amb}$	(T3, INT1) (Tamb, INT1)	DIFF
CE	$\Delta T_{Cool}$ $\Delta T_{Amb}$	DIV
Median CE 3	(CE, INT3)	ME

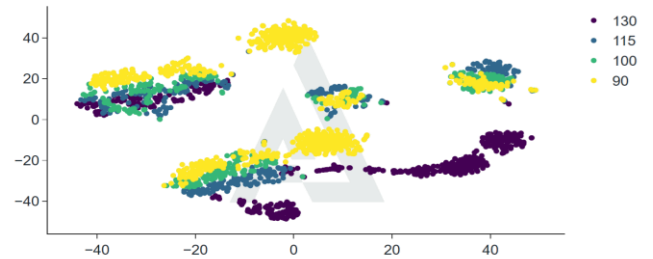
Cooler condition



Valve condition



Hydraulic accumulator / bar



Internal pump leakage

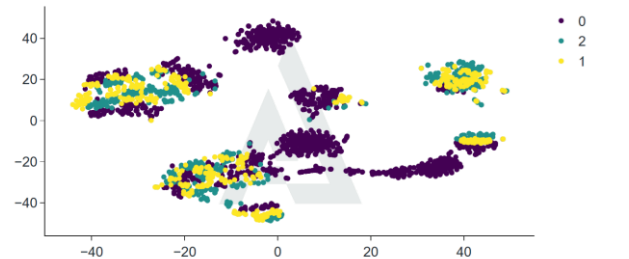


Figure 4. Visualization of an unsupervised, hybrid machine learning model developed and stored in the DeepFMEA framework. It was used for diagnostics of the hydraulic system dataset (Helwig, 2018).



As described in detail, the DeepFMEA framework also enables in quantifying and evaluating the risk reduction potential of developed Detection Methods. Even though Helwig 2015 does not provide the information needed to fully evaluate the risk reduction potential, we still give an outlook of how such an evaluation could be performed.

We introduce the following variables:

$R_F$ : The expected failure rate of a failure mode, i.e. the number of faults per asset per year

$C_F$ : Cost of downtime, including cost of spare parts and interventions needed for repair

$C_{DI}$ : Cost of a diagnostic intervention, needed to diagnose if a detected failure incident is an actual failure

$C_{PI}$ : Cost of a proactive intervention, needed to prevent an upcoming failure incident

$TPR, FPR, FNR$ : True positive rate, false positive rate and false negative rate of the developed detection method

We can then define the risk reduction potential as the difference between the total cost without implementing the detection method and the total cost with the detection method:

$$\text{Risk Reduction Potential} = R_F * C_F - R_F * (C_F * FNR + C_{DI} * FPR + (C_{DI} + C_{PI}) * TPR)$$

## 5. CONCLUSION: APPLICATION IN REAL-LIFE INDUSTRIAL SETTINGS

*DeepFMEA* introduces a novel abstraction to the design of data-driven PHM tools. It aims to:

- Enhance PHM development teams by incorporating best practices for systematically capturing and leveraging existing domain knowledge,
- Foster efficient collaboration among process experts, maintenance professionals, and data specialists in PHM projects,
- Streamline development workflows through standardization, which is essential for reusing specialized data management and MLOps modules. This standardization alleviates the burden of custom developments on PHM project budgets.

The application of this framework in real-world PHM projects across three equipment manufacturers has provided valuable insights:

- **Application-agnostic:** Tested in the steel, food & beverage, and machining industries, the framework's underlying data model has proven to be applicable for a broad range of similarly motivated use cases across different applications.

- **Model-agnostic:** Each implementation, whether employing simple dynamic thresholding techniques, classical ML algorithms and or advanced algorithms such as graph neural networks, benefited from *DeepFMEA*'s abstraction of the detection method by enabling the use of reusable MLOps modules.
- **Extensible:** The framework accommodated various initial requirements, from simple monitoring to advanced diagnostics, offering clear pathways for evolving PHM tools to increase the prescriptive value of their outputs as more data becomes available.
- **Flexible:** It could be implemented both in new (“green field”) projects and as an extension to existing data structures.
- **Automation:** Standardizing the capture and management of domain knowledge has facilitated the development of a graphical user interface, making PHM more accessible to non-data specialists, and reducing repetitive tasks for data specialists.

However, our experiences also highlight areas for improvement and expansion:

- **Risk Assessment Simplifications:** The current framework introduces strong simplifications for risk quantification and the assessment of the risk-reduction potential of PHM tools. It does not account for non-economic risks such as environmental and safety hazards. Moreover, it overlooks the multifaceted economic benefits of PHM, including waste reduction, energy efficiency, and productivity gains. A more refined model is necessary to accurately capture these aspects.
- **MLOps:** While *DeepFMEA* promotes standard MLOps practices for maintaining PHM tool trustworthiness, the unique challenges of PHM, such as limited benchmark data, organizational challenges in creating reliable data feedback loops to operators and maintainers, and varied operational contexts, demand more nuanced solutions. Addressing these challenges with innovative approaches should be a priority for the PHM research community and development teams to encourage broader adoption of data-driven PHM tools.

## References

- Arias Chao, M. & Kulkarni, C. & Goebel, K. & Fink, O. (2022), Fusing Physics-based and Deep Learning Models for Prognostics
- Basson, M. (2019). *RCM3.0: Risk-Based Reliability Centered Maintenance*, Industrial Press Inc. U.S.
- Battaglia, P. W. et al. (2018). Relational inductive biases, deep learning, and graph networks, *arXiv preprint arXiv:1806.01261*.
- Eker, O. F., Camci, F., & Jennions, I. K. (2016). Physics-based prognostic modelling of filter clogging phenomena. *Mechanical Systems and Signal Processing*, 75, 395–412. doi: 10.1016/j.ymsp.2015.12.011
- Ervural, B., & Ayaz, H.I. (2023). A fully data-driven FMEA framework for risk assessment on manufacturing processes using a hybrid approach. *Engineering Failure Analysis*, 152, 107525. doi: 10.1016/j.engfailanal.2023.107525
- Gassner, M., & Nilsson, J., & Nilsson, E., & Palmé, T. & Züfle, H. & Bernero, S.. (2014). A Data-Driven Approach for Analysing the Operational Behaviour and Performance of an Industrial Flue Gas Desulphurisation Process. *Computer Aided Chemical Engineering*. 33. Doi: 10.1016/B978-0-444-63456-6.50111-3.
- Hagemeyer, S., & Zeiler, P. & Huber, M. F. (2022), On the Integration of Fundamental Knowledge about Degradation Processes into Data-Driven Diagnostics and Prognostics Using Theory-Guided Data Science , *Proceedings of the 7th European Conference of the Prognostics and Health Management Society 2022* doi: 10.36001/phme.2022.v7i1.3352
- Helwig, N, & Pignanelli, E., & Schütze, A. (2015). Condition Monitoring of a Complex Hydraulic System Using Multivariate Statistics. *Proc. I2MTC-2015 - 2015 IEEE International Instrumentation and Measurement Technology Conference*, doi: 10.1109/I2MTC.2015.7151267
- Helwig, N, & Pignanelli, E., & Schütze, A. (2018). Condition monitoring of hydraulic systems. *UCI Machine Learning Repository*, doi: 10.24432/C5CW21.
- Kreuzberger, D., & Kühl, N., & Hirschl, S. (2023) Machine Learning Operations (MLOps): Overview, Definition, and Architecture, *IEEE Access*, vol. 11, pp. 31866-31879, 2023, doi: 10.1109/ACCESS.2023.3262138.
- Liao, L. & Köttig, F. (2016), A hybrid framework combining data-driven and model-based methods for system remaining useful life prediction, *Applied Soft Computing*, 44, 191-199, Doi: 10.1016/j.asoc.2016.03.013.
- Luo, W., Hu, T., Ye, Y., Zhang, C. & Wie, Y. (2020), A hybrid predictive maintenance approach for CNC machine tool driven by Digital Twin, *Robotics and Computer-Integrated Manufacturing*, 65, 101974, Doi: 10.1016/j.rcim.2020.101974
- Mantravadi, S., & Møller, C., & LI, C., & Schnyder, R. (2022). Design choices for next-generation IIoT-connected MES/MOM: An empirical study on smart factories. *Robot. Comput.-Integr. Manuf.* 73, C (Feb 2022). Doi: <https://doi.org/10.1016/j.rcim.2021.102225>
- Nahar, N., & Zhou, S., & Lewis, G., & Kästner, C. (2022). Collaboration challenges in building ML-enabled systems: communication, documentation, engineering, and process. *Proceedings of the 44th International Conference on Software Engineering (ICSE '22)*. Doi: <https://doi.org/10.1145/3510003.3510209>
- Nunes, P., Santos, J. & Rocha, E. (2023), Challenges in predictive maintenance – A review, *CIRP Journal of Manufacturing Science and Technology*, 40, 53-67, Doi: 10.1016/j.cirpj.2022.11.004
- Pothhoff, J. P., & Tienken, C., & Friedli, T. (2023), Business Models for Commercializing Predictive Maintenance in the Context of Industrial Sustainability, *Smart Services Summit. SMSESU 2022. Progress in IS. Springer, Cham*. doi: [https://doi.org/10.1007/978-3-031-36698-7\\_17](https://doi.org/10.1007/978-3-031-36698-7_17)
- Rai, R., & Sahu, C. K. (2020). Driven by data or derived through physics? a review of hybrid physics guided machine learning techniques with cyber-physical system (CPS) focus. *IEEE Access*, 8, 71050–71073. doi: 10.1109/access.2020.2987324
- Rausand, M., & Barros, A., & Hoyland, A. (2003), *System Reliability Theory: Models, Statistical Methods, and Applications*, 3<sup>rd</sup> Edition, Wiley
- Trauth, D., Bergs, T., Gülpen, C., Maaß, W., Mayer, J., Musa, H., Niemiets, P., Rohnfelder, A., Schaltegger, M., Seuttler, S., Starke, J., Szych, E., Unterberg, M. (2020) Monetarisierung von Fertigungsdaten, *Statusberichte aus der Produktionstechnik 2020*, 340-364, Doi: 10.24406/ipt-n-589615
- Ungvári, E., & Gyurika, I.G., & Csiszér, T. (2023). Evaluation of the failure effects of a screwing station using a new approached FMEA. *MethodsX*, 11, 102278. doi: 10.1016/j.mex.2023.102278
- Vollert, S., Atzmueller, M., & Theissler, A. (2021). Interpretable machine learning: A brief survey from the predictive maintenance perspective. In *Proceedings of the IEEE international conference on emerging technologies and factory automation (ETFA)* . Vasteras, Sweden: IEEE. doi: 10.1109/etfa45728.2021.9613467
- Wireman, T. (2004), *Total Productive Maintenance*, Industrial Press Inc. U. S.
- Zhao, H., & Wang, Y., & Duan, J., & Huang, C., & Cao, D., & Tong, Y., & Xu, B., & Bai, J., & Tong, J., & Zhang, Q. (2020). Multivariate Time-Series Anomaly Detection via Graph Attention Network, *2020 IEEE International Conference on Data Mining* doi: 10.1109/ICDM50108.2020.00093.
- Zio, E. (2022) Prognostics and Health Management (PHM): Where are we and where do we (need to) go in theory and practice, *Reliability Engineering & System Safety*, Volume 218, Part A, doi: 10.1016/j.ress.2021.108119