IAI DevOps: A Systematic Framework for Prognostic Model Lifecycle Management

Ziqi Guo, Tingwen Bao, Wenlong Wu, Chao Jin[1]
Industrial AI Lab & Department of Product R&D
CyberInsight Technology, Co, Ltd
Beijing, People's Republic of China
[1] chao.jin@cyber-insight.com

Jay Lee
Center for Intelligent Maintenance Systems (IMS)
University of Cincinnati
Cincinnati, OH, United States of America

Abstract—**This paper proposes IAI DevOps, a systematic framework to address challenges of developing and operationalizing AI models in manufacturing industries, with an emphasis on prognostics and health management (PHM) applications. The paper starts by introducing the growing need of accelerating AI model development and enhancing its lifecycle reliability in industrial AI systems. The framework for industrial AI (IAI) DevOps is then proposed, with key components including industrial data management, streamlined model training, risk monitoring and model update, and decision support and feedback. After detailed introduction of the function and principles of each component, an implementation of the supporting platform is illustrated for the reference of the readers. With the help of the IAI DevOps framework, the authors believe that interdisciplinary engineering teams can collaborate better to accelerate the implementation of PHM systems as well as broader industrial AI applications in manufacturing industries.**

Keywords-Industrial AI; DevOps; model lifecycle management; PHM

I. INTRODUCTION

Under the flourishing trend of digital transformation in manufacturing, prognostics and health management (PHM) has become a leading use case of industrial AI in recent years [10]. Prognostic models enable the analysis of machine data in real-time and predict potential failure and/or remaining useful life (RUL) of the monitored machine/component. Insights generated by prognostics models can be used to help operators schedule maintenance activities proactively and improve overall operation efficiency.

Although methodologies related to the development of prognostic models have been actively researched in numerous industries (wind energy [18-21], CNC milling [8][23-24], etc.), limited applications have been successfully launched in production and deployed at scale or achieved desired return-of-investment (ROI) [1][21-22][25]. The authors believe that it is mainly because of lacking in training data and large environment variances [2-3][15]. To address this issue, Javed *et al.* [2] provide a comprehensive review of state-of-art prognostics algorithms and their applicability under different system conditions. They also discuss a framework of technology readiness level (TRL) that can be used to understand the maturity of prognostics. Moreover, Sankararaman [15] dives into the issue of uncertainty management, with an emphasis on remaining useful life prediction. He interprets uncertainty from a Bayesian perspective and proposes computational approaches for uncertainty quantification and propagation. Previous works like these lay the groundwork for our discussion as well as motivate the need of a general engineering framework for industrial AI applications.

In this paper, we propose IAI DevOps, a framework for prognostic model lifecycle management, with the objective of shortening the "go-to-market" cycle of industrial AI applications while handling real-world uncertainties appropriately. To construct the framework, we borrow key concepts from DevOps, a methodology in software engineering for continuous delivery (CD) of a system, and reinvent them in the context of industrial AI applications.

The rest of the paper is organized as follows: Section II analyzes the engineering challenges in developing PHM applications and further illustrates the need for model lifecycle management; Section III presents the architecture of the IAI DevOps framework and its design principles in detail; Section IV gives an implementation of its supporting platform; Section V concludes the paper by summarizing its contribution and areas for future study.

II. ENGINEERING CHALLENGES IN DEVELOPING PROGNOSTICS AND HEALTH MANAGEMENT APPLICATIONS

According to a survey published by PwC in 2018 [1], only 11% of the 268 surveyed manufacturing companies had successfully launched predictive maintenance in-house, which is lower than market expectation. In general, there are 3 major challenges impeding engineers from developing PHM applications in real world:

- **Shortage of appropriate training data**: Industrial data in real world suffer from "3B" challenges, i.e. bad quality, broken, below-surface [3]. Not only does it require significant efforts to conduct data cleaning and preprocessing, but the training data is often

978-1-7281-0860-5/19/$31.00 ©2019 IEEE

insufficient, especially the amount of failure labels. Unlike most consumer Internet applications, where the volume of training data is often at a high level, data for prognostic model training is usually small. It causes barriers especially for developers without domain knowledge to overcome the cold-start stage.

- **Large real-world uncertainty**: In manufacturing, reliability and robustness of a system is paramount. However, prognostic models largely suffer from unmanaged real-world uncertainties, such as, a working regime shift of monitored machines, an update in production recipe, a change in environment conditions, or even human impacts, etc. While building a model that is validated under all possible conditions is desired, it is very unlikely to completely avoid uncertainties. When encountering variance or unknown regimes, the model needs to reduce the possibility of firing false alarms while being able to adapt to new situations. This would require constant updates or rebuilding of the model, while its engineering cost adds constrains to the problem.
- **Difficulty in interdisciplinary collaboration**: The successful delivery of a PHM application requires interdisciplinary collaboration from hardware engineers, data engineers, data scientists, software engineers, and domain experts (DT/AT/PT/OT, namely data technology/analytics technology/platform technology/operation technology) [14]. The prognostic model has to be integrated into the final system in an agile and repeatable manner, in order to deliver and operationalize the model in production.

As a result, the challenges lead to a long and sometimes uncontrollable development cycle of PHM applications. Also, due to limited adaptability of the prognostic model, undesirable risks are added into production and undermine customer confidence. To address these challenges, the IAI DevOps framework is proposed in the next section to systematically manage major the entire lifecycle of a prognostic model, which enables efficient model development and comprehensive risk management.

III. ARCHITECTURE OF THE IAI DEVOPS FRAMEWORK

A. Overview

DevOps is commonly defined as a set of software development practices that combine development (Dev) and operations (Ops) to shorten the delivery cycle while ensuring high quality of the software [11-13]. Compared with DevOps in software engineering, industrial AI (IAI) DevOps focuses on managing the lifecycle and uncertainties of AI models. As shown in Figure 1, an IAI DevOps cycle starts with model architecture design, which is supported by a thorough understanding of field requirements and domain knowledge. Based on that, additional data acquisition hardware may be installed. Once training data is prepared, data scientists will perform exploratory data analysis, model training, and validation. After offline performance criteria are met, the model can be integrated into the software and deployed in the production environment, which marks the transition to the "Ops" phase. The model will then predict on real-time data to provide insights to the users. At the same time, its performance will also be continuously monitored. In case of an abnormal event, such as performance deterioration, data regime shift, user requirement changes, etc., the model will re-enter the "Dev" phase for further update. Compared with ad-hoc model training practices, Table I lists the goals that engineers hope to achieve through IAI DevOps.

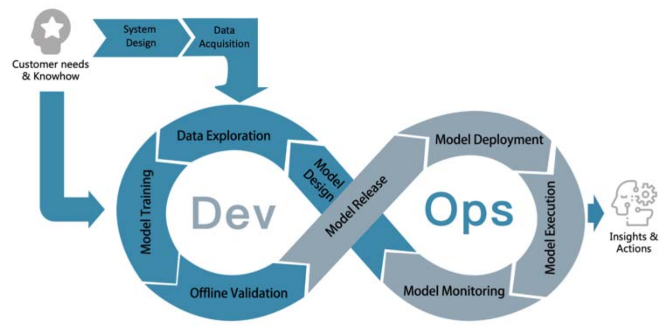

Figure 1. The Industrial AI (IAI) DevOps cycle

TABLE I. OBEJECTIVES OF IAI DEVOPS

Stage	Objectives
Model Development (Dev)	• **Streamlined:** Model development process is streamlined where standard building blocks are connected together to form a complete training pipeline. • **Reusability:** Training pipelines are built in a reusable manner. • **Traceability:** Every update or modification of the model and its corresponding metainformation can be traced.
Model Operation (Ops)	• **Seamless integration:** Models can be easily integrated into the software system with few errors or flaws. • **Risk management:** Model status can be constantly monitored once deployed in the production environment to ensure reliable output and timely model updates. • **Interpretability:** Model outputs can be well-explained to end-users through domain-informed visualization to support near real-time decision and feedback. • **Self-adaptive:** Models are expected to semi-automatically evolve to enable efficient model updates.

In order to achieve the above objectives, the following framework is proposed, with key components including industrial data management, streamlined model training, risk monitoring and model update, and decision support and feedback. Examples and tactics are also introduced in corresponding sections for algorithmic implementation in the future.

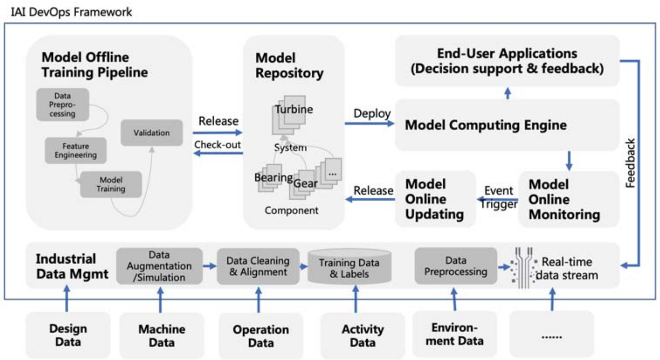

Figure 2. Architecture of the IAI DevOps Framework

B. *Industrial Data Management*

The industrial data management module is responsible for providing appropriate data for model training and computation. Leaving aside the mechanisms for big data storage, it mainly contains 4 sub-parts: data augmentation, data integration, data cleaning, and data quality management. Data augmentation addresses the shortage of training data via minority oversampling or data simulation. Data integration is to align data with different frequencies, formats and sources (e.g. machine controller data, add-on sensor data, maintenance log, manufacturing execution system (MES) data, etc.) together to create the desired data stream for model development. Data cleaning is to remove nulls and outliers in input data, which utilizes rule-based or distribution-based algorithms. Data quality management addresses the quality aspect as data being transmitted and transformed throughout the data chain. Gateways may be set after different stages of data processing (e.g. data integration, feature extraction) to avoid invalid data from further proceeding. General data quality inspection guidelines are given in [16], while a specific example for wind turbine vibration data quality check is listed in Table II [3][17]. Note that a high-degree of automation in this module would be vital to eliminate redundancy and reduce input uncertainty in the long run.

TABLE II. EXAMPLE OF QUALITY CRITERIA FOR WIND TURBINE VIBRATION DATA [3][17]

Check Method	Data Processing	Check Value	Threshold
Mean check	Mean value of vibration signal	Mean value	Smaller than 1e-5 (should be decently small)
RMS check	RMS value of vibration signal	RMS value	1e-5–0.05 (minimum energy rule and dynamic range rule)
Parseval's theorem-based Energy conservation rule	Time domain RMS and frequency domain RMS level should be close (conservation of energy for FFT)	RMS(x(t)) − RMS (X(f))	Smaller than 0.1%
Statistical distribution rule	Fit normal distribution of vibration signal	Hellinger-like distance and Komogorov distance of empirical and fitted distribution	<0.12 for K-distance <0.1 for H-distance
N-point rule	Number of unique points in the vibration signal	N-point	Depends on sampling frequency (<1 in this case due to very high sampling frequency)
U-point check	Number of unique points in the vibration signal	Portion of unique points to the length of dataset	>99.99% in this case
Positive and negative point check	Portion of positive and negative points to the length of dataset	Max[P(+), P(−)]	<52% for this case (the value should be close to 50%)
Derivative check	Derivative of vibration signal	RMS value of derivative signal; number of derivative value that exceeds threshold	0.015 for RMS derivative

Due to the large volume of data collected from machine, it is preferred that portions of this module are to be distributed to the edge to decrease communication bandwidth. Certain features can be first computed at edge devices and then sent back to the cloud for integrative analysis.

C. *Streamlined Model Training*

Similar to an assembly line, industrial AI models should be developed in a streamlined manner to achieve higher engineering efficiency. Instead of having one person doing all the work, different engineering roles, including data engineers, data scientists and machine experts, will work together to contribute to the consummation of the final model. To better decouple the work and standardize internal collaboration, the authors propose a streamlined model training pipeline in the form of directed acyclic graphs (DAG), as shown in Figure 3. Every node in the graph is a pre-defined building block with reconfigurable parameters and is supplied by different domain experts in the team. Data scientists who are in charge of the final delivery will connect selective building blocks together to form the model training pipeline.

In the example of a high-speed railway bearing diagnostics, a data engineer developed modules that stitch high-frequency and low-frequency data together and conduct basic data cleaning; a vibration data expert developed noise reduction and signal processing algorithms; a machine learning engineer provided multiple condition clustering modules. Under the guidance of the leader of the team, a junior data scientist was able to create several training pipelines with the building blocks and conduct experiments without intensive training. Modules and pipelines can also be saved and reused in the future.

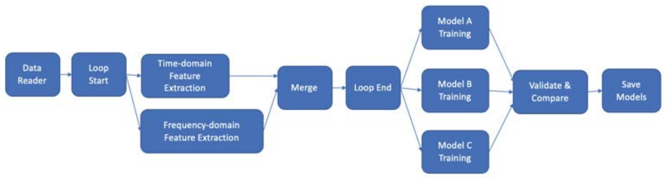

Figure 3. Sample Model Training Pipeline

D. *Model Risk Monitoring*

Due to uncertainty in industrial systems and their operating conditions, one should not expect to have an AI model that works well forever. As illustrated in Figure 4, the context in which the model predicts is likely to evolve over time. To reduce false outputs, continuous risk monitoring will be required throughout the lifecycle of the model.

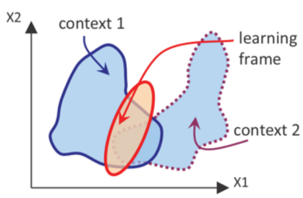

Figure 4. Illustration of Variation in Model Operating Context [2]

For PHM models, the following metrics can be used for a relatively comprehensive evaluation of model performance and uncertainty in production [7].

TABLE III. CLASSIFICATION OF PROGNOSTICS MODEL METRICS [7]

Category	Metric
Algorithm Performance	• Accuracy • Precision • Robustness • Convergence
Model Uncertainty	• Regime shift • Confidence level
Computational Performance	• Success rate • Time & complexity • Memory & IO
Cost-benefit-risk	• ROI • Maintenance cost & savings

In particular, monitoring machine operating regimes – such as loading, machine status (production mode, working receipt, etc.), and environment factors (cabin temperature, air density, etc.) – can be crucial for industrial AI models. When a regime shift occurs, models may switch to a more conservative prediction mode (e.g. rule-based) or output a preset code to warn about this to avoid false alerts. If an accumulative regime shift is detected online, developers should be notified to update the model.

In addition, this module also monitors the occurrence of operational events that potentially break model assumptions, such as maintenance activities, machine parameter calibration, and new product introduction (NPI). When these events are identified, the module will notify the developers to re-evaluate the suitability of the model.

E. *Continuous Model Updating*

It is necessary for industrial AI models to be continuously updated to maintain their reliability. Model updating can be classified into two scenarios: one is model structural change, which means re-designing most or all parts of the model; the other is non-structural change, in which case only model (hyper-)parameters are re-optimized using new data and experience.

In the second case, model updating is hopeful to become semi-automated. Alarms from the risk monitoring module or user feedback would trigger the start of a model update. Recent data and labels will be queried and previous training pipelines will be rerun with new configurations from the developers or the users. In the case of a machine tool cutting tool remaining useful life (RUL) prediction, for instance, a model update is performed after a new work recipe is employed. During the update, the machine operator is presented with diagrams of raw vibration data and top trendy features automatically selected by the training pipeline. The operator only needs to inspect if any major anomaly exists. After the confirmation of the operator, the pipeline will extract cutting data under the new recipe, refit model parameters, and conduct validation. After another user inspection on the validation results, the model will be automatically deployed online. By enabling mechanisms like this, model updating can be finished within factory sites and the cost for model maintenance will be curtailed as well. To ensure reliability, developers will take over if any manual or automatic inspection fails to pass.

Algorithm-wise, several research works have proposed methods for semi-automatic model updating in specific scenarios. For example, Liu et al proposed an adaptive clustering method for fault diagnosis in railway PHM applications with self-organizing maps (SOM) [4]. Tang et al utilized the strength of Bayesian networks and presents an inner-outer loop updating mechanisms for model hyperparameters [5]. Wang presented a trajectory-based RUL prediction algorithm (TSBP) for jet engines that enables sample-based updates [6].

In the long run, model structural change will be inevitable, since initial training data and system know-how are insufficient. As data and label accumulate, new sensors get installed, and studies on the system mechanism deepen, the model will be refined to perform better through different evolving routes. In the case of wind turbine PHM, a model evolving roadmap is presented in Figure 5 as an example. A series dimensions will be considered in making such transitions, such as data conditions (ratio/availability of faulty data), model requirements (interpretability, scalability, computational complexity), degree of domain-know (knowledge/representation into failure mechanism, knowledge/representation into system dynamics), etc. References [2][8][9] list more exhaustive criteria for this issue.

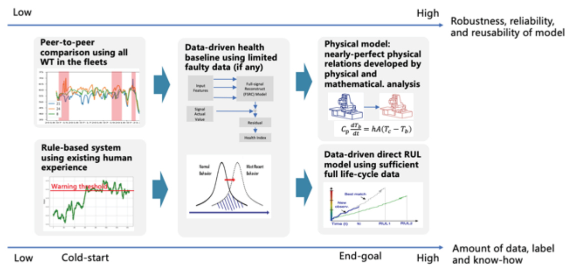

Figure 5. Prognostics Model Roadmap of Wind Turbine Generator Bearing Fault Prediction [2]

F. *Decision Support & Feedback Interface*

A user interface is required for closed-loop decision support and execution feedback. Information presented on the interface should be illustrative for end-users (e.g. maintenance operators) to understand why the model gives certain alarms and suggestions. As shown in Figure 6, instead of a single alarm level, the interface also shows recent risk trends, diagrams of key model features, diagrams of corresponding raw data. Sometimes the decision path of the model (from component to system) will also be presented for users to understand its outputs. In this way, the model results are seamlessly cross-referenced with the operators' perceptions, avoiding obvious false positives.

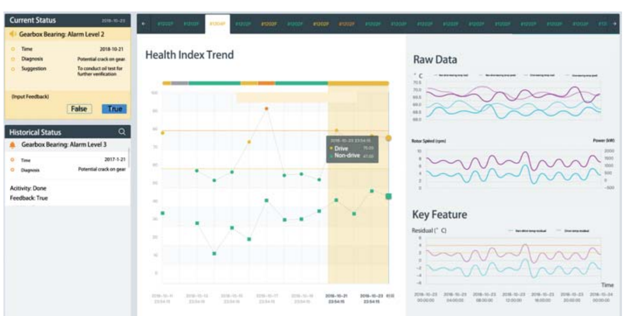

Figure 6. Visualization of a Prognostics Model Output for Wind Turbine Generator Bearing

Also, on-site inspection results are collected simultaneously as feedbacks into the system, which can be used for label generation or constructing a maintenance knowledge base. The knowledge base stores fault characterization and maintenance procedures along with data snapshots to support future maintenance activities as well as subsequent model update.

IV. Implementation of the IAI DevOps Platform

Figure 7 presents the architecture of a supporting platform of IAI DevOps. Functions of each component are introduced as follows.

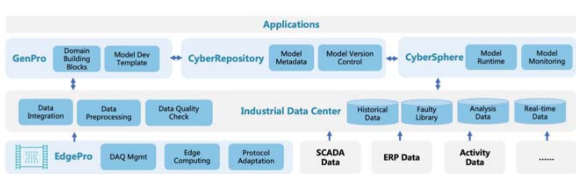

Figure 7. Architecture of the IAI DevOps Platform

1) *EdgePro: Edge Computing & IoT Management*

EdgePro is an edge-computing module that can be integrated into data acquisition (DAQ) devices to perform data processing and model computation at the edge. It is aimed at ensuring timely responses and reducing transmission pressure in places with poor connectivity. EdgePro also serves as the management tool for DAQ devices with functionalities such as remote reconfiguration of DAQ parameters, edge model update, etc.

2) *GenPro: Reusable Model Training*

GenPro is an integrated development environment (IDE) that contains modularized operators and modeling templates for rapid model building and updating. Operators are algorithms and methods developed by domain experts or data scientists to solve fundamental problems in different prognostic stages, such as signal preprocessing, feature extraction and selection, baseline model training, model validation, etc. One can directly connect those building blocks together to form a complete model training pipeline. Additionally, when scaling up or conducting updates, the pipelines can be reused. The user interface of GenPro is presented in Figure 8.

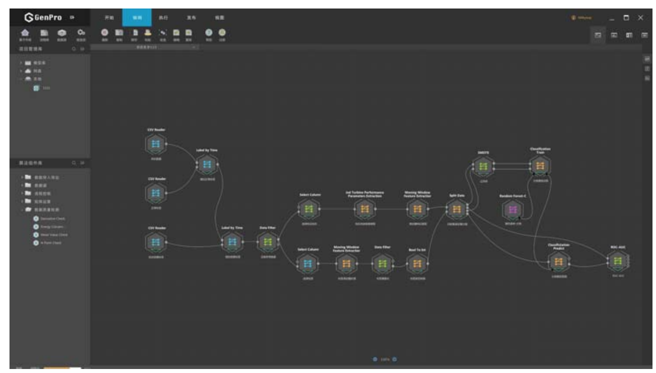

Figure 8. User Interface of the IAI DevOps Platform

3) *CyberSphere: Model Validation, Execution, and Monitoring*

CyberSphere is a computational engine for prognostic models. Before the model is deployed into production, CyberSphere uses offline data to validate its correctness; after the model is deployed online, CyberSphere executes it with user-defined strategies and continuously monitors its performance. It also serves as a centralized data processing layer that preprocesses input data and checks its quality. With CyberSphere's unified operating environment, consistency between offline and online model execution is ensured.

4) *CyberRepository: Model Management*

CyberRepository is a model repository that stores different versions/variants of a prognostic model throughout its lifecycle. Access control is also enabled for model approvals. Models stored in CyberRepository are crucial for model traceability and knowledge accumulation.

The platform has been used to deliver several industrial AI applications in different manufacturing sectors, as shown in Table IV.

TABLE IV. DIFFERENT USE CASES OF THE IAI DEVOPS PLATFORM

Use Case	Characteristics
Wind turbine predictive health management	• Large fleet of machines with a lot of components to be monitored. This results in hundreds of prognostics models being scaled and operated on the platform simultaneously.
CNC machine tool RUL prediction	• High-frequency vibration data with edge computing applied for data cleaning and feature extraction. • Semi-automatic model updating is applied to accommodate frequent changes in working receipts.
Intelligent energy dispatch in factory	• Data from a variety of management systems are collected and centralized for model computing. Human feedbacks are collected upon every energy adjustment for model improvement and risk monitoring.

V. CONCLUSION & FUTURE WORK

This paper introduces IAI DevOps, a systematic framework for prognostics model lifecycle management. It addresses the engineering challenges of developing and operationalizing AI models in traditional industries, with an emphasis on PHM applications. By illustrating the architecture of the framework and the design principles for its component, the authors wish to offer the readers a high-level understanding of IAI DevOps. Indeed, PHM applications can be different from case to case, but the philosophy for tackling the engineering aspect of the problem can be comparable. With the IAI DevOps framework, we hope to accelerate the development and deployment of industrial AI models that eventually bring values to end users. Going forward, the authors will dive into each component of the framework and study its detailed technical schemes in the context of representative industrial scenarios. Areas with higher research priorities are data quality inspection, incremental learning algorithms, automatic maintenance knowledge mapping, etc.

ACKNOWLEDGMENT

The authors would like to thank other colleagues from Innovation Center and the Department of Product R&D, for their dedication and hard-work in accelerating the landing of AI applications in manufacturing industries, including but not limited to Dr. Wenjing Jin, Huimin Deng, Xiaoming Wang, Wei Fu, Rui Gan, Lei Zhang, Zhongnan Yan, Fang Chen, and Dr. Zongchang Liu.

REFERENCES

[1] M. Haarman *et al.*, “Beyond the hype: PdM 4.0 delivers results,” PricewaterhouseCoopers B.V. and Mainnovation, 2018.

[2] K. Javed, R. Gouriveau, and N. Zerhouni, “State of the art and taxonomy of prognostics approaches, trends of prognostics applications and open issues towards maturity at different technology readiness levels,” *Mechanical Systems and Signal Processing*, vol. 94, pp. 214–236, Sep. 2017.

[3] J. Lee, C. Jin, Z. Liu, and H. Davari Ardakani, “Introduction to Data-Driven Methodologies for Prognostics and Health Management,” *Probabilistic Prognostics and Health Management of Energy Systems*, pp. 9–32, 2017.

[4] Z. Liu *et al.*, “Industrial AI Enabled Prognostics for High-speed Railway Systems,” in *2018 IEEE International Conference on Prognostics and Health Management (ICPHM)*, 2018.

[5] L. Tang, G. J. Kacprzynski, K. Goebel, and G. Vachtsevanos, “Methodologies for uncertainty management in prognostics,” in *2009 IEEE Aerospace conference*, 2009.

[6] T. Wang, “Trajectory Similarity Based Prediction for Remaining Useful Life Estimation,” 2010.

[7] A. Saxena *et al.*, “Metrics for evaluating performance of prognostic techniques,” in *2008 International Conference on Prognostics and Health Management*, 2008.

[8] J. Lee, F. Wu, W. Zhao, M. Ghaffari, L. Liao, and D. Siegel, “Prognostics and health management design for rotary machinery systems—Reviews, methodology and applications,” *Mechanical Systems and Signal Processing*, vol. 42, no. 1–2, pp. 314–334, Jan. 2014.

[9] J. Z. Sikorska, M. Hodkiewicz, and L. Ma, “Prognostic modelling options for remaining useful life estimation by industry,” *Mechanical Systems and Signal Processing*, vol. 25, no. 5, pp. 1803–1836, Jul. 2011.

[10] J. Zhou et al., “Data-Driven Transformation: Global Industrial Internet Platform Case Study,” [10] China Industrial Control Systems Cyber Emergency Response Team, 2018.

[11] Atlassian, “What is DevOps?,” Atlassian, 2016. [Online]. Available: http://www.atlassian.com/devops/.

[12] “What is DevOps?,” Amazon Web Services, Inc., 2019. [Online]. Available: http://aws.amazon.com/devops/what-is-devops/.

[13] “DevOps for Hybrid Cloud: an IBM Point of View,” IBM, 2017.

[14] J. Lee, H. Davari, J. Singh, and V. Pandhare, “Industrial Artificial Intelligence for industry 4.0-based manufacturing systems,” Manufacturing Letters, vol. 18, pp. 20–23, Oct. 2018.

[15] S. Sankararaman, “Significance, interpretation, and quantification of uncertainty in prognostics and remaining useful life prediction,” *Mechanical Systems and Signal Processing*, vol. 52–53, pp. 228–247, Feb. 2015.

[16] Y. Chen, F. Zhu, and J. Lee, “Data quality evaluation and improvement for prognostic modeling using visual assessment based data partitioning method,” *Computers in Industry*, vol. 64, no. 3, pp. 214–225, Apr. 2013.

[17] A. Jabłoński, T. Barszcz, and M. Bielecka, “Automatic validation of vibration signals in wind farm distributed monitoring systems,” *Measurement*, vol. 44, no. 10, pp. 1954–1967, Dec. 2011.

[18] P. Cambron, A. Tahan, C. Masson, and F. Pelletier, “Bearing temperature monitoring of a Wind Turbine using physics-based model,” *Journal of Quality in Maintenance Engineering*, vol. 23, no. 4, pp. 479–488, Oct. 2017.

[19] E. Lapira, D. Brisset, H. Davari Ardakani, D. Siegel, and J. Lee, “Wind turbine performance assessment using multi-regime modeling approach,” *Renewable Energy*, vol. 45, pp. 86–95, Sep. 2012.

[20] M. Schlechtingen, I. F. Santos, and S. Achiche, “Wind turbine condition monitoring based on SCADA data using normal behavior models. Part 1: System description,” *Applied Soft Computing*, vol. 13, no. 1, pp. 259–270, Jan. 2013.

[21] H. D. M. de Azevedo, A. M. Araújo, and N. Bouchonneau, “A review of wind turbine bearing condition monitoring: State of the art and challenges,” *Renewable and Sustainable Energy Reviews*, vol. 56, pp. 368–379, Apr. 2016.

[22] W. Yang, P. J. Tavner, C. J. Crabtree, Y. Feng, and Y. Qiu, “Wind turbine condition monitoring: technical and commercial challenges,” *Wind Energy*, vol. 17, no. 5, pp. 673–693, Aug. 2012.

[23] T. Benkedjouh, K. Medjaher, N. Zerhouni, and S. Rechak, “Health assessment and life prediction of cutting tools based on support vector regression,” *Journal of Intelligent Manufacturing*, vol. 26, no. 2, pp. 213–223, Apr. 2013.

[24] D. A. Tobon-Mejia, K. Medjaher, and N. Zerhouni, “CNC machine tool’s wear diagnostic and prognostic by using dynamic Bayesian networks,” *Mechanical Systems and Signal Processing*, vol. 28, pp. 167–182, Apr. 2012.

[25] M. Schallehn, C. Schorling, P. Bowen, and O. Straehle, “Beyond Proofs of Concept: Scaling the Industrial IoT,” Bain& Company, 2019.