Data Generation with a Physical Model to Support Machine Learning Research for Predictive Maintenance

Patrick Klein and Ralph Bergmann

Business Information Systems II
University of Trier
54286 Trier, Germany
[kleinp][bergmann]@uni-trier.de,
http://www.wi2.uni-trier.de

Abstract. Today, manufacturing machines are continuously equipped with various sensors, whose data enable to derive a comprehensive picture of the current state of each machine. Predictive maintenance approaches make use of this data in order to predict the occurrence of possible failures before they actually occur, thereby significantly reducing production and service costs. The application of machine learning to sensor data streams is an essential part of data-driven predictive maintenance in order to find the patterns in the data that are indicators of upcoming faults. Thus, research on machine learning for predictive maintenance is a recent and very challenging field. However, there are currently no appropriate data sets available that can be used for this kind of research. In this paper we therefore propose an approach for the generation of predictive maintenance data by using a physical Fischertechnik model factory equipped with several sensors. Different ways of reproducing real failures using this model are presented as well as a general procedure for data generation.

Keywords: Data Generation · Machine Learning · Predictive Maintenance · Industry 4.0.

1 Introduction

Today, we are facing the beginning of a transformation towards the fourth industrial revolution, called Industry 4.0, which is characterized by manufacturing and service innovations based on cyber-physical systems, big data and an intensive use of methods from artificial intelligence [15]. In particular, manufacturing environments are equipped with various sensors and actuators that are fully connected for the purpose of building an industrial Internet of Things, with the aim of using data in real time for decision making. An important field of service innovation is related to diagnosis and maintenance of manufacturing machines. For this purpose, manufacturing machines are equipped with various sensors, whose data enable to derive a comprehensive picture of the current state of each machine. Based on this data, occurring problems can be diagnosed and more importantly, upcoming problems can be predicted prior to their occurrence. In particular, predictive maintenance (PredM) aims at foreseeing a breakdown of the system to be maintained by detecting early signs of failure in order to make maintenance work more proactive. It has been adopted by various sectors in manufacturing and service industries in order to improve reliability, safety, availability, efficiency and quality as well as to protect the environment [20].

For PredM to work, knowledge is required about characteristic data patterns that are indicators of specific faults that have occurred or that are likely to occur in the future. Due to the large number of potential faults as well as the large variety of production machinery and components used, the manual definition of such patterns is not feasible. Instead, machine learning (ML) is required to automatically derive such patterns from available data. Patterns could describe healthy states as well as states which are in the transition phase towards an upcoming breakdown. The application of ML to PredM, however, comes along with a variety of challenges, in particular related to the complexity and heterogeneity of data, the lack of labelled data, the need for transfer learning, as well as the necessity of deriving models that enable explainable decision support. Thus, PredM is an ideal application area for research in ML due to its complexity and practical relevance. However, appropriate and realistic data is required to conduct such research. While plenty of data sets are available for research purposes in ML, there is a lack of data that can be used immediately for PredM applications. Also it is nearly impossible (at least for Universities) to get real data from industry due to the serious confidentiality issues involved.

In this paper we therefore address the issue of obtaining data appropriate for ML research in PredM. The contribution of this paper is manyfold. First, we present a brief overview of PredM and the involved research challenges for ML (Sect. 2). Then, we characterize the required data to address these challenges, analyze existing data sets as well as methods for the generation of new research data (Sect. 3). The main contribution of the paper is the presentation of an approach for the generation of PredM data based on a physical model of a specific production environment implemented based on a Fischertechnik (FT) model factory (Sect. 4). We further describe various ways for injecting faulty behaviour in a defined way into the FT model factory, in order to collect the respective data that can be used to learn the related patterns for prediction. We describe the current state of realization as well as our planned future work (Sect. 5).

2 Predictive Maintenance and Machine Learning

2.1 Predictive Maintenance

Industrial maintenance involves all measures that are required to ensure or to re-establish the proper functioning of industrial machinery. The goal is to prevent the occurrence of failures that could lead to breakdowns or downtimes of machines or that could lead to safety concerns. For example, a common cause of failure is wear, which is a gradual damage or deformation of material due to forces, which occurs in many mechanical components, in particular in bearings, O-rings, or gears. In general, systematic maintenance procedures increase the availability of machines, reduce costs, and enable to schedule required maintenance actions. Traditional, preventive maintenance involves the systematic inspection of machines following a fixed time schedule or a fixed mileage, which is based on the simplified assumption that failures mostly occur after a certain and known operating time or effort. However, it often happens that failures occur before the scheduled maintenance activity or that maintenance actions are performed although they are not yet necessary. Thus, PredM aims at overcoming the fixed time schedule approach by introducing methods that are able to individually predict upcoming failures. The goal is to perform maintenance actions only when they are really necessary, i.e., not too early and not too late. For companies, PredM has the advantage that maintenance costs can be reduced significantly by better utilization of capacities and by avoiding downtimes in manufacturing.

PredM is based on forecasting failures based on the current state captured by various sensors, such as vibration, temperature, humidity, or acoustic sensors. In addition, parameters characterizing the current state in the production process (e.g. position sensors or switches as well as the activity state of actuators) are relevant. Machines in real production environments may have hundreds of various sensors producing data streams with high frequency. As there is a wide variety of factories, plants, and other systems that depend on reliability, it is not economically feasible to build a customized solution for each plant. Thus, research is focusing on universal parts of machines, such as bearings, gearboxes, motors, valves, pumps, compressors, and so on to find universal solutions.

2.2 Machine Learning for Predictive Maintenance

The increasing number of sensor data streams makes manual monitoring and analysis impossible, which is why ML and especially deep learning are suitable for PredM data processing [13, 25]. They are mostly applied to the typical PredM tasks [11] such as Remain Useful Life (RUL), Root Cause Analysis also referred to as Fault Diagnosis (FD), Fault Prediction (FP), and Maintenance Strategy Optimization (MSO). The prediction of RUL values for components is probably the most prominent application which is a regression task with multivariate time series as input, however, sometimes it is performed as a classification task in which the RUL values are discretized from larger ranges to classes. For instance, Babu et al. [4] applied a convolutional neural network for RUL prediction and Yuan et al. [23] compare different recurrent neural network architectures for RUL and FD of an aircraft turbofan engine. Furthermore, FP is used to predict upcoming incorrect functioning which is not caused by wear, for instance, a future incorrect positioning of a robot arm in a manufacturing process or to predict defects on a production line [24]. Finally, MSO supports decision-making in questions when, what, and how an upcoming maintenance task should be carried out.

2.3 Research Challenges

These tasks lead to a number of challenges and requirements for the application of ML methods in PredM. The first issue is the large number of distributed, heterogeneous sensor data streams with non-uniform time stamps. It is difficult to determine the subset of relevant data streams for the detection of a failure as well as the time frame in which these data streams produce characteristic patterns that are an indication of this failure. Quite often it is difficult to label correctly the occurrence of a certain failure, as maintenance protocols are usually the only source of information about when which failure has occurred. This leads to huge problems related to the data preparation prior to the use of ML algorithms. In addition, failures are usually the exception, which makes the data sets highly unbalanced. Although the overall volume of data is huge, the number of different failure cases for a certain type of failure is rather small. This leads to the need for transfer learning, in order to be able to transfer a learned failure model from one machine component to a different, but similar component. Finally, the ability to explain a certain prediction is also very important in PredM in order to enable a human operator to assess and verify an automatically proposed maintenance action. All these different topics make PredM an interesting field of application for ML research.

3 Research Data for Machine Learning in Predictive Maintenance

3.1 Requirements on Research Data for Predictive Maintenance Research

For conducting ML research for PredM (as well as for other kinds of Industry 4.0 applications of ML), it is necessary to have data available that is to some degree comparable to the data in industrial settings. Thus, data sets are necessary which are composed of various data streams with different characteristics (according to the type of sensors used in industry) together with related data about the current status of the production component or process. For learning to predict failures, the data streams must contain patterns, similar to those that occur in real situations, for example as a result of wear. Also the data sets must be at least partially labelled with the respective fault to be predicted. Ideally, we need large data sets describing several instances of the same fault and data sets describing the same fault in various different but similar components (e.g. wear in the bearings of different motors) to investigate transfer learning approaches.

3.2 Existing Data Sets

Eker et al. [5] benchmarked six common run-to-failure data sets for their application to data-driven prognostics and found only two of them applicable. One of them [22] consists of only 68 run-to-failure measurements and the other one [19] consists of between 100 and 260 train examples for four different settings and failure types. The major sources for industrial prognostic data sets are the NASA Prognostics Data Repository with 16 data sets[1] as well as a collection consisting of a dozen data sets from previously organized data competitions by Prognostic and Health Management Society[2] (of engineering systems).

The available data sets known to us have a variety of issues which makes them not suitable for ML research on PredM. The data sets are either too small for ML, several of them have anonymized features, and nearly all sets are insufficiently labelled for the mentioned PredM tasks. Further, the data sets usually only consider individual components or working station cells, but do not provide a comprehensive snapshot of a factory's sensor data.

3.3 Approaches for Data Generation

Since there is no sufficient or adequate data from real industrial factories publicly available for research purposes, it is desirable to collect or generate them. Sensor data generation without the real production environment at hand can be categorized into four groups: 1. fully synthetical, 2. synthetical based on previous data, 3. synthetical based on a virtual simulation model, and finally 4. based on a simplified physical model.

Fully Synthetic Data Generation Fully synthetic data generation means that sensor data is generated by an algorithm based on given parameters. The resulting streams are based on a statistical structure and can contain concept drifts (changing of underlying statistical properties over time). Typical parameters are the data generating distribution (e.g. Gaussian), noise rate, data dimensionality, and generation periodicity. For instance, Hahsler et al. [9] provide a software framework for generation and analysis of fully synthetical data streams.

Synthetic Data Generation Based on Previous Data Another way to generate sensor data is to learn the underlying properties of an existing data distribution in order to generate new data. This can be done by training a generative and discriminative neural model by learning either explicitly the parameters of the distribution [2] or implicitly with a generative adversarial network for time series [7].

Synthetic Data Generation Based on a Virtual Simulation Model A further approach is the creation of a virtual simulation model with the properties of the real model and use this for data generation. This approach, for example, has been applied to aircraft gas turbines [19] and to create a virtual factory [12] including detailed machine level data streams for testing machine health data analytics applications.

[1] https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/
[2] https://www.phmsociety.org/

Data Collection Based on a Simplified Physical Model Instead of using a virtual model of a factory or machine, there is also the possibility of using a real but simplified physical model. Regarding the level of abstraction and the constituents of the model, such models can be divided into two categories.

The first category are models which are equipped with real industrial components leading to minor abstractions. Examples of such factories are Learning Factories [1] such as AutFab [21] or the SmartFactory[3] particularly established for Industry 4.0 research. Also small physical models for the generation of specific faults, such as bearing faults [18] exist.

The second category consists of models with a higher level of abstraction, which are build using non-industrial components. The advantage of this approach is the significantly low cost involved in building such a model. There are several platforms which enable the simple cost-efficient construction of such models. Among them the most popular are Lego Mindstorms[4] and Fischertechnik (FT)[5]. Examples are the Smart-LEGO Factory[6] at DFKI, the FT plant model implemented for applicability validation of Industry 4.0 components [16] and a FT punching workstation built to demonstrate how a generic client can access data generated from the workstation [3].

4 Generating Research Data for Predictive Maintenance by FT Model Factory

As publicly available data sets for research on ML for PredM are rare and limited to single components or working cells, we now describe an approach for the generation of such data based on a simplified physical model. The aim is to provide a cost-effective way to generate data according to the requirements sketched in Sect. 3.1. This requires constructing a physical model of a factory, attaching appropriate sensors and related data collection hard- and software, as well as developing means for simulation faults.

4.1 A Physical Model Factory for Data Generation

Our Industry 4.0 factory model is built based upon the FT Factory Simulation[1] as shown in Fig. 1. It has been selected due to its superior robustness compared to Lego Mindstorms. As FT is also used in University education for automation engineers, the available FT components are already closer to real industrial components than those of Lego. The FT factory model that we use consists of four modules: a sorting line with color detection, a multi-processing station with oven and milling machine, a high-bay warehouse, and a vacuum gripper

[3] http://www.smartfactory.de/
[4] https://www.lego.com/en-us/mindstorms
[5] https://www.fischertechnik.de/en
[6] https://www.dfki.de/web/aktuelles/dfki-cebit-2016/smart-lego
[1] https://www.fischertechnik.de/en/service/elearning/simulating/fabrik-simulation-24v

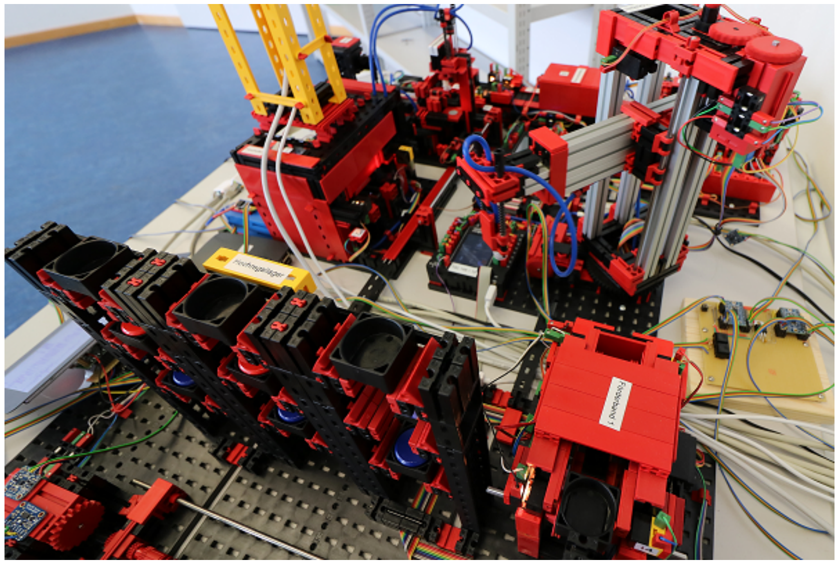

Fig. 1. The FT Factory Simulation Model

robot. Each module is operated by its own controller based on an ARM Cortex A8 CPU with various analog and digital input/output ports running under a LINUX kernel. Overall, the model consists of nine light barriers, ten switches, twelve motors and three compressors. Moreover, we enhanced the model with six three-axis acceleration sensors that are mounted on motors and compressors for vibration measuring and four differential pressure sensors are measuring the pressure generated from the three compressors. These sensors are connected to a separate Raspberry Pi controller. Furthermore, two micro-electro-mechanical systems (MEMS) each with a gyroscope, an accelerometer, and a geomagnetic sensor will be installed on the Robotic Vacuum Gripper and the High-Bay's storage and dispensing machine. RFID-Tags for product identification and a reading device are also planned. Further, we will extend the oven model with a heating pad in order to change the color of thermo-colored product materials. This process will be monitored by a thermal imaging camera.

All controllers are connected via an Ethernet network and communicate via remote procedure calls. The overall control software for the entire production process is distributed over the controllers, each of which is in charge of a certain module of the factory. For processing the generated data, we selected the SMACK stack [8] as a Lambda architecture [10] implementation because it is often used for Big Data applications in industry. Thus, we set up each controller as a producer to the high throughput distributed messaging system Apache Kafka [14]. Apache Cassandra was installed as a database for batch processing and we further plan to use Apache Spark for stream processing and ML research.

The overall manufacturing process is designed as a cycle, meaning that data can be generated without manual interference. The process starts from the High Bay where workpieces are dispensed and transported to the Multi Processing Station. After processing, they are sorted by color, transported by the Robotic Vacuum Gripper and finally stored in the High Bay where the process repeats.

4.2 Reproduction of Failures

By using the FT model along with the developed software, the manufacturing process is executed in a continuous loop. As FT blocks are quite robust and all physical connections are very stable, problems occur quite rarely and hence the model is able to run properly over a very long period of time. However, in order to be able to produce data for predictive maintenance, faults must occur such that the resulting data can be collected. As such faults do not occur naturally (within an acceptable time limit) realistic faults must be artificially infused into the model.

Figure 2 describes the interplay between reality, our FT model, the creation of faults, and finally the data generation. In general, reality defines which failure types are measurable and reasonable. Our FT model is a smaller and simplified representation of reality and due to this it certainly restricts our ability to reproduce realistic defects. Also life expectancy for components in real machines is months to years and degradation processes are very slow. Thus we have to compress the time dimension, i.e., we have to significantly shorten the time during which a certain type of fault causes its typical effects. Based on these limitations we define plausible defects that can be simulated by our physical model such that data is generated that can be used for learning and evaluating prognostic models on predictive maintenance.

In general, there are several ways in which behaviour can be generated similar to a failure in reality.

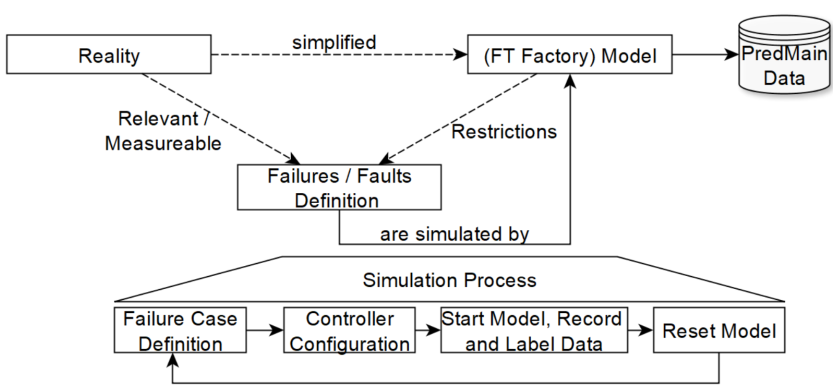

Fig. 2. Methodology and process for reproduction of failures.

Amending the Model by Additional Actuators In order to produce an abnormal behaviour of the physical model, additional actuators can be integrated whose activities cause certain disturbances. For example, workpieces can be pushed from the conveyor, a pressure line can be virtually broken by inserting a pressure valve, or an additional motor can be inserted to produce an additional mechanical load on a drive shaft.

Adapting the Controller Software for Actuator Based on knowledge about how certain failures (e.g. motor problems due to wear) have an impact on an actuator (reduced or unstable revolution speed), the controller software can be designed such that it controls the actuator in a way that it behaves as if it would exhibit the failure. For example, the motor supply voltage can be reduced following a certain pattern or the frequency for the pulse-width modulation of motor power supply can be lowered to increase vibration.

Simulating Defective Sensors Faults related to a defect of any kind of sensor are also quite likely. They could also have a significant impact on the production process, in case the sensor is used within the control procedure of the machine. For example, a defective position switch might cause problems, as a gripper is not able to adjust itself to the correct position. Defective sensors can be easily simulated as part of the control software by manipulating the value they produce.

4.3 Data Generation Process

The just described ways of generating faulty behaviour have to be embedded into an overall generation process for maintenance data. Therefore, the following data generation process has been developed, allowing to generate a large number of labelled maintenance data sets automatically. This process runs in a loop consisting of four steps (see Fig. 2):

1. Selection of the particular error (e.g. motor failure due to wearing) to be produced in the current run, including the relevant parameters (which motor, degree of wearing, failure pattern curve, time horizon of wear process, etc.).
2. Configuration of the controller software to run the factory in a mode, in which the respective fault reproduction is enabled.
3. Start of the controller software to run the production process. During the run of the factory, all data is collected and stored in an Apache Cassandra data base and labelled with the respective failure being produced.
4. After the failure has occurred, the factory model is reset to a defined initial state compensating for any inconsistencies that might have resulted from the insertion of the failure.

4.4 Example Case

In reality, bearing faults are the biggest failure source with almost 40% to 50% of electrical motors. They result in vibration signatures of higher amplitudes, increased noise, and also in a reduced motor torque and thus motor speed [17]. Bearing faults do not occur suddenly, but develop over time. To simulate this kind of failure in the FT model, we first run the model in the regular mode (to collect data unaffected by faults) and then we slowly reduce the motor speed over time and in addition decrease the frequency for the pulse-width modulation of motor power. Depending on which motor is affected, the reduced speed, for example, leads to an increased duration of the movings of the gripper or increases the time for transport of the workpiece on the conveyor. This leads to longer delays until the respective signals from position switches arrive. In addition the acceleration sensors record an increased vibration amplitude on one axis. This data is recorded along with the data of all other sensors of the factory model. Besides the sensors directly affected by the reduced motor speed, other sensors might also be affected as an indirect consequence of the reduced speed. After several runs of the data generation process with different variations of the failure parameters, appropriate data is available for learning failure patterns and to address the research challenges of ML for PredM.

5 Conclusion and Future Work

In this paper, we address the problem of data generation to enable ML research for PredM. We surveyed currently available data sets and present several approaches for data generation. We then present an new approach for PredM data generation based on a FT factory model. As of today, the mechanical and electrical side of the model is nearly completely realized, the sensor data is collected, processed, and stored using the SMACK-Stack as described. First failures, as the example case just described, are implemented.

Future work will first of all address the implementation of additional failure scenarios based on the approaches described in Sect. 4.2. This work is quite difficult as it requires at least a basic understanding of typical mechanical faults and their consequences in order to be able to reproduce them on the model. In general it would be desirable to find ways of validating to which degree the data we are collecting is realistic compared to real production data. However, this will be clearly difficult as data from comparable real components, such as provided in [6] for rolling bearings, is hardly available. However, we assume that for the development of ML methods for PredM the exact reproduction of patterns from real industrial factories is not required, as the goal of ML methods is to find patterns according to the production environment at hand. Thus, we are confident that the developed FT factory model is an appropriate means to perform laboratory research on ML in a well controlled environment. We also plan to publish the gained data sets such that they could be used by other researchers as well.

References

1. Abele, E., Metternich, J., Tisch, M., Chryssolouris, G., Sihn, W., ElMaraghy, H., Hummel, V., Ranz, F.: Learning factories for research, education, and training.

Procedia CiRp **32**, 1–6 (2015)
2. Alzantot, M., Chakraborty, S., Srivastava, M.: Sensegen: A deep learning architecture for synthetic sensor data generation. In: Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on. pp. 188–193. IEEE (2017)
3. Angione, G., Barbosa, J., Gosewehr, F., Leitão, P., Massa, D., Matos, J., Peres, R.S., Rocha, A.D., Wermann, J.: Integration and deployment of a distributed and pluggable industrial architecture for the perform project. Procedia Manufacturing **11**, 896–904 (2017)
4. Babu, G.S., Zhao, P., Li, X.L.: Deep convolutional neural network based regression approach for estimation of remaining useful life. In: International conference on database systems for advanced applications. pp. 214–228. Springer (2016)
5. Eker, Ö.F., Camci, F., Jennions, I.K.: Major challenges in prognostics: study on benchmarking prognostic datasets (2012)
6. El-Thalji, I., Jantunen, E.: Fault analysis of the wear fault development in rolling bearings. Engineering Failure Analysis **57**, 470–482 (2015)
7. Esteban, C., Hyland, S.L., Rätsch, G.: Real-valued (medical) time series generation with recurrent conditional gans. arXiv preprint arXiv:1706.02633 (2017)
8. Estrada, R., Ruiz, I.: Big data smack. Apress, Berkeley, CA (2016)
9. Hahsler, M., Bolaos, M., Forrest, J.: Introduction to stream: An extensible framework for data stream clustering research with r. Journal of Statistical Software, Articles **76**(14), 1–50 (2017). https://doi.org/10.18637/jss.v076.i14, https://www.jstatsoft.org/v076/i14
10. Hausenblas, M., Bijnens, N.: Lambda architecture, http://lambda-architecture.net/
11. Hegedűs, C., Varga, P., Moldován, I.: The mantis architecture for proactive maintenance. In: 2018 5th International Conference on Control, Decision and Information Technologies (CoDIT). pp. 719–724. IEEE (2018)
12. Jain, S., Shao, G., Shin, S.J.: Manufacturing data analytics using a virtual factory representation. International Journal of Production Research **55**(18), 5450–5464 (2017)
13. Khan, S., Yairi, T.: A review on the application of deep learning in system health management. Mechanical Systems and Signal Processing **107**, 241–265 (2018)
14. Kreps, J., Narkhede, N., Rao, J., et al.: Kafka: A distributed messaging system for log processing (2011)
15. Lee, J., Kao, H.A., Yang, S.: Service innovation and smart analytics for industry 4.0 and big data environment. Procedia CIRP **16**, 3 – 8 (2014). https://doi.org/https://doi.org/10.1016/j.procir.2014.02.001, http://www.sciencedirect.com/science/article/pii/S2212827114000857, product Services Systems and Value Creation. Proceedings of the 6th CIRP Conference on Industrial Product-Service Systems
16. Lüder, A., Schleipen, M., Schmidt, N., Pfrommer, J., Henßen, R.: One step towards an industry 4.0 component. In: Automation Science and Engineering (CASE), 2017 13th IEEE Conference on. pp. 1268–1273. IEEE (2017)
17. Nandi, S., Toliyat, H.A., Li, X.: Condition monitoring and fault diagnosis of electrical motors a review. IEEE transactions on energy conversion **20**(4), 719–729 (2005)
18. Nectoux, P., Gouriveau, R., Medjaher, K., Ramasso, E., Chebel-Morello, B., Zerhouni, N., Varnier, C.: Pronostia: An experimental platform for bearings accelerated degradation tests. In: IEEE International Conference on Prognostics and

Health Management, PHM'12. pp. 1–8. IEEE Catalog Number: CPF12PHM-CDR (2012)
19. Saxena, A., Goebel, K., Simon, D., Eklund, N.: Damage propagation modeling for aircraft engine run-to-failure simulation. In: Prognostics and Health Management, 2008. PHM 2008. International Conference on. pp. 1–9. IEEE (2008)
20. Selcuk, S.: Predictive maintenance, its implementation and latest trends. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture **231**(9), 1670–1679 (2017)
21. Simons, S., Abé, P., Neser, S.: Learning in the autfab–the fully automated industrie 4.0 learning factory of the university of applied sciences darmstadt. Procedia Manufacturing **9**, 81–88 (2017)
22. Virkler, D.A., Hillberry, B., Goel, P.: The statistical nature of fatigue crack propagation. Journal of Engineering Materials and Technology **101**(2), 148–153 (1979)
23. Yuan, M., Wu, Y., Lin, L.: Fault diagnosis and remaining useful life estimation of aero engine using lstm neural network. In: Aircraft Utility Systems (AUS), IEEE International Conference on. pp. 135–140. IEEE (2016)
24. Zhang, D., Xu, B., Wood, J.: Predict failures in production lines: A two-stage approach with clustering and supervised learning. In: Big Data (Big Data), 2016 IEEE International Conference on. pp. 2070–2074. IEEE (2016)
25. Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., Gao, R.X.: Deep learning and its applications to machine health monitoring: A survey. arXiv preprint arXiv:1612.07640 (2016)