Moving Maintenance from Preventative to Predictive with ML

How Machine Learning Is Changing Maintenance Regimens

By Stuart Cording (Elektor)

Keeping industrial systems operational is like spinning plates – you're kept busy trying to avoid any falling and breaking. While preventative maintenance regimens, optimally implemented, work very well, critical equipment must still be taken offline regularly for servicing. Additionally, wastage due to replacing parts and lubricants earlier than necessary is costly and bad for the environment. Predictive maintenance is increasingly being used to optimize this essential function, ranging from static, dynamic, and statistical analysis to the use of machine learning to prognose pending failure promptly. The overall goal: to maximize uptime and reduce maintenance costs.

The cost of keeping manufacturing equipment operational and the potential perils should it go wrong have long been documented. There is a famous story in engineering circles of the ex-employee who is asked to return to fix an obstinate machine [1]. Upon resolving the issue, an itemized bill is requested. The repair action (tapping with a hammer or marking the area of the failure with a chalk cross) makes up a mere hundredth of the total sum charged. The remainder is for knowing where to tap or apply the chalk cross.

For those who have worked in manufacturing, and perhaps still for some today, a scheduled week's closure for maintenance, requiring almost everyone to take a week's holiday, was not unusual. The absence of employees and workpieces allowed regular maintenance to be undertaken, with machines being stripped-down and rebuilt, pipes to be cleaned, and visual inspections to be completed. It is, however, a little absurd that an entire plant needs to be closed for a week to avoid the impact a potential machine breakdown may have.

According to Behrtech [2], unplanned downtime at industrial manufacturers costs $50 billion per year, with a single downtime incident costing around $30 k to $50 k per hour (**Figure 1**). Tangible outcomes include loss in production capacity, underutilized labor costs, depletion in inventory, and delivery delays. On the intangible side, ever-pending machine failure can impact morale, dampening innovation, place stress on employees, and, longer-term, lead to lost customers due to a damaged reputation.

Most organizations employ preventative maintenance schedules, not dissimilar to the approach taken with annual vehicle servicing. Klüber Lubrication, a manufacturer of industrial lubricants, offers software solutions for keeping on top of maintenance tasks [3]. Furthermore, measurement of industrial equipment's energy consumption coupled with lubricant condition analysis is also offered. This enables machine lifetime to be increased with a resultant reduction in energy consumption and lubricant employed.

Of course, the ultimate goal is to gather all the available data generated at an industrial plant or complex and use it to search proactively for pending equipment failure. This is, after all, part of the promise of Industry 4.0 and the Industrial Internet of Things (IIoT). The challenge is that, in most plants, the sensors and controllers are put in place to control processes with the goal of manufacturing something. If a heated tank has a temperature and liquid volume sensor, the system can only detect if the heater is broken, a valve is stuck, or a pump defective. At this point, the equipment is already broken.

Continuous monitoring of temperature data during the heating process can be used to detect a pending heater fault. By differentiating the data to determine the rate of heating, any change in the heater's functionality could be detected by defining acceptable heating-rate limits. Other failures may require the addition of more sensors to catch them. Electric valves may require current sensors that allow abnormal changes in current drawn to be determined. Pumps could benefit from vibration sensors that can detect pending bearing failure.

The maintenance engineer's crystal ball

The pinnacle of maintenance engineering is to be able to determine a pending failure or detect a servicing need. Such a predictive maintenance approach allows unplanned downtime to be reduced and enables improved scheduling of planned downtime. Industrial heavyweight Bosch Connected Industry offers a solution in this domain with their Nexeed Industrial Application System [4]. Consisting of an array of packages that can be confectioned as required, it acquires production and machine data, visualizes it in a harmonized output dashboard, and allows users to define rules that indicate operation outside expected norms.

OSRAM in Berlin has based their Ticket Manager on this platform to use the predictive maintenance insights it can deliver. The site is steeped in history, known for its hand manufacture of incandescent bulbs during the 1920s and innovation in every step of the lamp manufacturing process [5]. Today, the site is responsible for the Xenarc range of headlamps. Thanks to the long-standing and intense focus on quality, the existing continuous improvement process (CIP) had reached the point of diminishing returns [6].

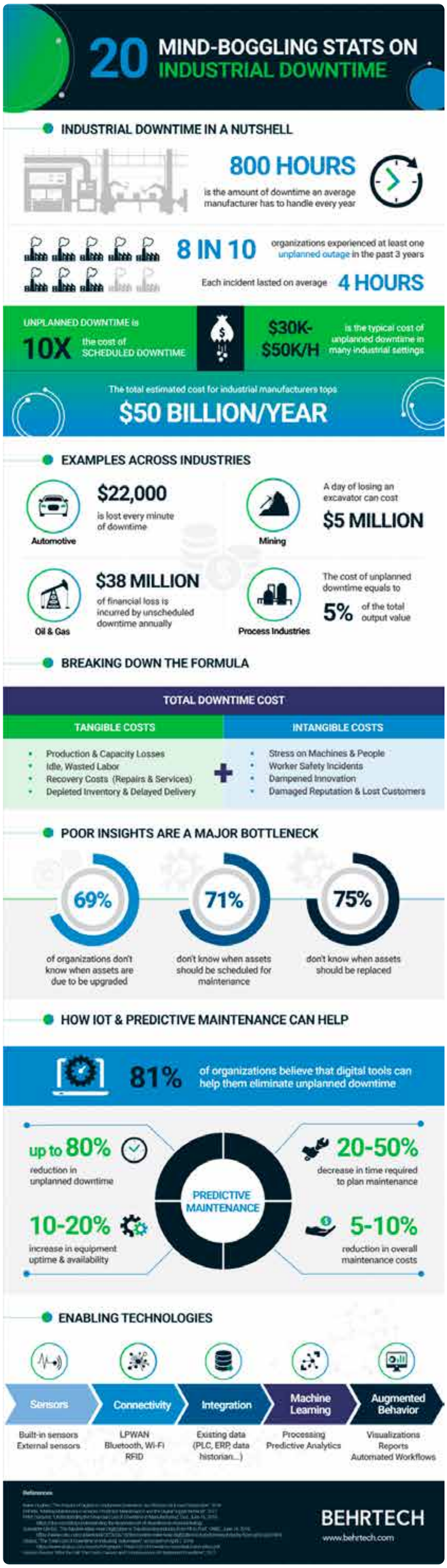

Figure 1: Industrial manufacturing downtime is estimated to cost $50 billion per year, and around $22k per minute in the automotive industry. (Copyright: BehrTech [2])

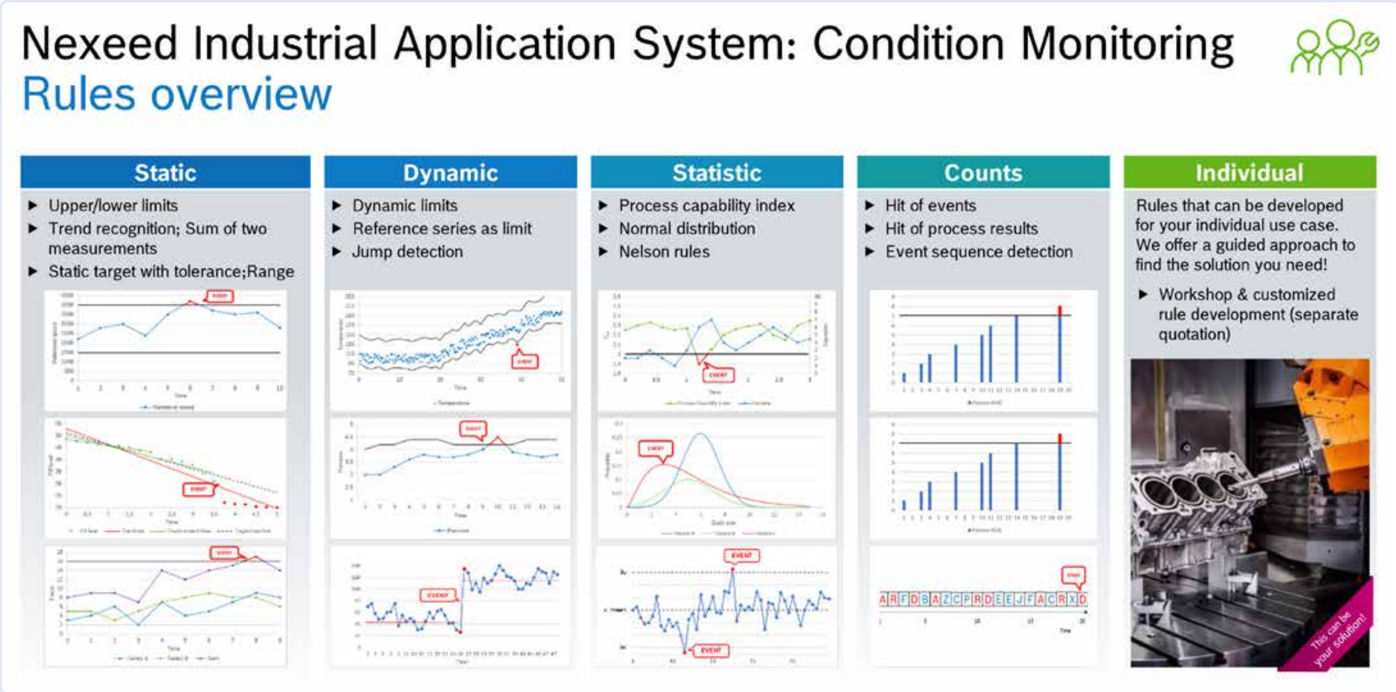

Figure 2: Static, dynamic, and statistical condition monitoring rules can be configured in Nexeed Industrial Application System to implement predictive maintenance. (Copyright: Bosch)

By linking their manufacturing equipment with Nexeed, they could further improve their manufacturing and maintenance processes. The platform delivers the data required in a format useable by the maintenance team while providing them the autonomy to program limits that, when crossed, trigger the issuance of maintenance tickets. The tickets are automatically prioritized, accessible via handheld devices, and directed to the engineer most appropriate for the task. Trigger rules can be simple, based on static limits or trend recognition, dynamic, detecting unexpected jumps in process values, or even use statistical analysis (**Figure 2**). If necessary, more complex rules can be defined according to individual use cases.

The impact of ML-supported maintenance

The mass networking of industrial complexes for Industry 4.0 suddenly makes them highly suitable for new data analysis approaches. Machine learning (ML) techniques are ideal for analyzing the millions of data points captured every day. However, the application of ML requires significant skill that starts with reviewing the suitability of the data.

ML requires vast amounts of data to determine the features that make up wanted and unwanted behavior. For example, to teach an algorithm to recognize zebras, you start by showing it many zebras images. However, during testing, it is likely to misclassify other black-and-white patterned animals, such as white tigers, lemurs, badgers, and skunks, as zebras. Therefore, you also need to provide images of these animals and tell the model that they do not represent zebras. This combination of valid and invalid examples improves the accuracy of the ML algorithm. Such datasets primarily contain data on functioning systems within industrial complexes, especially if they are well maintained. This makes it challenging to teach the algorithm the signs of equipment deterioration.

The next hurdle is labeling. Process data is typically collected from sensors with only the units attributed to the data, such as temperature, speed, and pressure. It is, therefore, unclear whether the data is related to correct or incorrect system or process operation.

A final challenge is a lack of data points due to missing sensors or disparities in data quantity from one sensor compared to others. Returning to our heated tank example, the process may deliver a temperature data point every second, but merely an open/closed status for valves and on/off for the pumps. Ideally, a pump requires vibration, temperature, and current sensors to detect anomalies in operation caused by lack of lubricant or excessive wear of bearings. Alternatively, data from which this information could be inferred (such as fill times) needs to be generated.

There is then a range of approaches available for applying ML [7]. Calculation of the remaining useful life (RUL) can be used to determine when equipment is likely to fail, which helps with maintenance scheduling. This could be in terms of hours of operation for machinery that runs continuously or cycles for aircraft turbines. Alternatively, anomalous behavior can be flagged in real-time using an incoming data stream, known as time series analysis. Another approach is to support failure analysis by using ML to suggest possible issue mitigation, such as pump rather than valve replacement, when a failure occurs.

RUL can be applied in different ways [8]. One approach is binary classification to answer the question "Will my machine fail next week?" to which we expect a yes/no answer. Alternatively, multiple classification can be used, providing yes/no answers to the question related to different periods of time (failure in 2, 4, or 6 weeks). ML models that can determine regression can be developed relatively quickly using available open-source tools such as scikit-learn [9], programmed in Python. To aid in the development process, organizations such as NASA's Prognostics Center of Excellence provide repositories of labeled data suited to developing ML failure prediction models.

One useful dataset is the Turbofan Engine Degradation Simulation Data Set [10] (C-MAPSS dataset). It provides data from 21 sensors with sensor noise, as would be apparent in actual data. Each dataset ends with engine failure, known as a run-to-failure dataset. RUL values are also included, allowing ML algorithm developers to compare their prediction models with reality. Each simulated engine also starts with a varying but acceptable level of wear, thus adding to the dataset's realism.

In a documented project [11] developing an RUL model by doubleSlash Net-Business, Munich, the confusion matrix shows their prediction algorithm correctly determined the turbine's breakdown in the next 30 days 63 times, but incorrectly determined a pending breakdown 12 times (**Figure 3**). Bearing in mind the safety implications of a non-functional turbine on an aircraft, the accuracy achieved can be considered to be very good. Furthermore, the model did not predict a no-failure case when a failure resulting from a defect would have occurred. In another project [12] analyzing the same C-MAPSS data with scikit-learn and H2O.ai [13], removal of data from specific sensors from the dataset improved root-mean-square error (RMSE). This highlights the critical role data scientists play during the development of ML models, as not all data in the dataset has equal impact or value on the quality of the model's output.

Such ML-supported predictive maintenance is already in use in the airline industry. Lufthansa has seen a 20% reduction in downtime [14] by using RapidMiner's data science platform [15]. The implementation benefits from the readily available data collected through existing preventative maintenance programs.

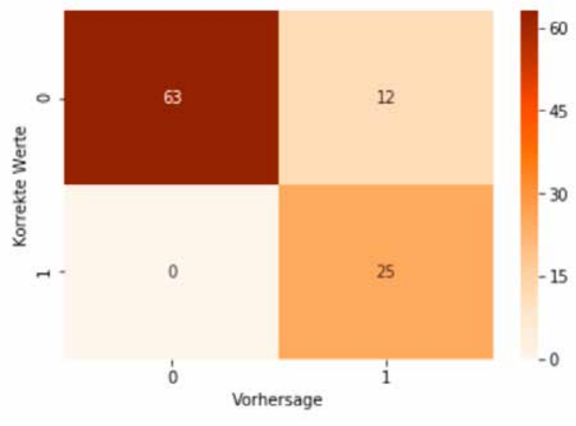

Figure 3: The confusion matrix resulting from the predictive failure analysis ML algorithm for a turbofan engine. (Copyright: doubleSlash Net-Business GmbH)

Of course, not everyone is versed in programming languages such as Python, and they may prefer a point-and-click approach. Talend [16] has developed a demonstrator that analyses accelerometer data from mobile handsets, then classifies it as resting, walking, or running. Based upon their enterprise big data platform, it accepts the streamed accelerometer data in real-time through a Kafka [17] message queue, then analyses it using a Random Forest model from Spark MLlib [18]. The results are displayed in a dashboard [19]. While not a predictive maintenance example, it does highlight an alternative implementation approach.

ML at the edge

One of the great industry concerns around ML is control over the data the algorithms rely upon. As a result, cloud-based ML implementations are often held in low regard, with solutions being sought that can be operated in-house instead. This is a challenge, as the execution of ML algorithms is a notoriously processor-intensive task. This is the reason why there are so many cloud-based artificial intelligence platforms on offer.

Solutions such as RapidLab's Edge AI Gateway [20] provide a solution. The term edge indicates that the processing capability sits on-site beside the equipment generating the data instead of offsite in the cloud. Not only does this ensure that data does not leave the building, it is often the only way to implement the real-time, low-latency ML support needed. Based upon the NVIDIA Jetson Nano system-on-module (**Figure 4**), it can be integrated directly with sensors on existing industrial networks (PROFIBUS, IO-Link, CAN). It also features two MIPI CSI-2 connectors for interfacing with cameras, another potential additional source of visual or thermal data for predictive maintenance algorithms.

Figure 4: Low-latency ML is enabled by solutions such as RapidLab's Edge AI Gateway featuring NVIDIA's Jetson Nano. (Source: rapidLAB.io)

While holistic analysis of industrial equipment for predictive maintenance purposes is one approach, the alternative is to build it into the equipment directly. Intelligent motors, valves, and pumps could use ML to determine their degree of wear using highly integrated sensors and software suited to operation on microcontrollers. One candidate for this approach is tinyML [21] [22], targeting operation within a 1 mW power envelope and kilobytes of random access memory (RAM). Another is AIfES (Artificial Intelligence for Embedded Systems), developed by the Fraunhofer Institute for Microelectronic Circuits and Systems. The advantage of this C-language library is that the training is undertaken on the microcontroller, rather than developing the model on a more powerful processing platform. To date, simple applications have been built to classify air gestures and handwritten numbers using an Arduino UNO [23] (**Figure 5**).

Figure 5: The AIfES from Fraunhofer Institute for Microelectronic Circuits and Systems runs handwriting recognition on an Arduino UNO. (Copyright: Fraunhofer IMS)

From preventive to predictive

It is clear that advancements in data science and ML can be combined with ongoing Industry 4.0 efforts. However, one outstanding challenge is how to acquire data suitable for training ML models. With existing sensors being used primarily for process control rather than equipment monitoring coupled with the absence context labeling, it is challenging to teach an ML algorithm the difference between regular operation and a pending failure.

Using accurate models to generate run-to-failure datasets, such as those shared by NASA on turbofan operation, is one option. However, these require significant investment, and, unlike aircraft engines, industrial systems are diverse, making it challenging to define standard, all-encompassing models. Perhaps the real solution lies with embedding ML, such as tinyML, into individual pumps, valves, and larger systems, like conveyers and robot arms. Existing maintenance platforms, such as Nexeed, could then undertake out-of-band monitoring of each component's ML assessment of wear level, rather than inferring it via a complex collection of networked sensors. What is clear is that, if applied, ML has a significant positive role to play in predictive maintenance applications.

(210264-01)

Questions or Comments?

If you have any technical questions about this article, feel free to contact the author by email at stuart.cording@elektor.com.

WEBLINKS

[1] "Knowing Where To Tap," Quote Investigator, March 2017: https://bit.ly/2OGQe9o
[2] "20 Mind-Boggling Stats on the Cost of Industrial Downtime," BehrTech: https://bit.ly/3dY4cfz
[3] KlüberEfficiencySupport webpage: https://bit.ly/3mHcXie
[4] Nexeed Industrial Application System webpage: https://bit.ly/2QjNNKx
[5] A. Burghart, B. Müller, W. Hanseder, "100 years of OSRAM - Light has a name," OSRAM GmbH, December 2006: https://bit.ly/3teHzdg
[6] "A Guide to Robotics and Automation," Distrelec (Page 76): https://bit.ly/3mJGtUG
[7] A. Gonfalonieri, "How to Implement Machine Learning For Predictive Maintenance," Towards Data Science, November 2019: https://bit.ly/32dv0D2
[8] J. Mehne, "So setzen Sie Predictive Maintenance richtig um," Computerwoche, January 2021: https://bit.ly/3teHTZE
[9] scikit-learn website: https://scikit-learn.org
[10] Turbofan Engine Degradation Simulation Data Set: https://go.nasa.gov/3uMH5M4
[11] J. Mehne, Predictive Maintenance Jupyter Notebook, GitHub, September 2019: https://bit.ly/3mJE6RO
[12] R. Alwis, S. Perera, "Machine Learning Techniques for Predictive Maintenance," InfoQ, May 2017: https://bit.ly/3mIBy67
[13] H2O.ai website: https://www.h2o.ai/
[14] "Using Machine Learning for Predictive Maintenance," RapidMiner, September 2020: https://bit.ly/2RrJY6p
[15] RapidMiner website: https://rapidminer.com/
[16] N. Krupa, "Applying Machine Learning to IoT Sensors," Talend, April 2017: https://bit.ly/2PQ15P5
[17] Kafka website: https://kafka.apache.org/
[18] Apache Spark MLLib website: https://spark.apache.org/mllib/
[19] "Analytics Dashboard - Talend IoT Demo," Talend, April 2017: https://bit.ly/3dcm5Z2
[20] RapidLab Edge AI Gateway: https://rapidlab.io/edge-ai-gateway/
[21] tinyML Foundation website: https://www.tinyml.org/
[22] C. J. Abate, "What Is Tiny Machine Learning (tinyML)?," Elektor, February 2021: https://www.elektormagazine.com/articles/tiny-machine-learning
[23] "Machine learning for sensors," Fraunhofer IMS, June 2019: https://bit.ly/3mJFzri