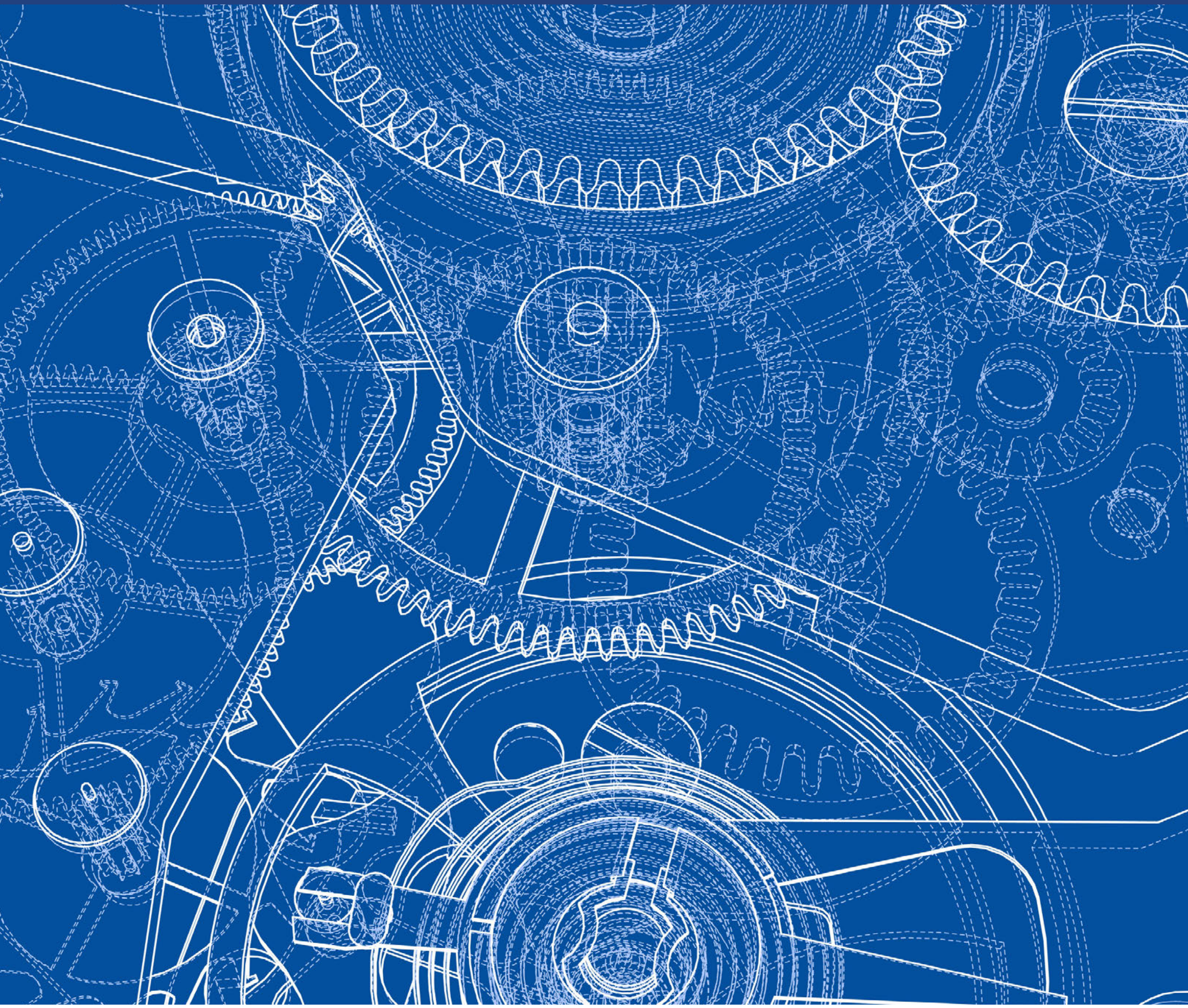
WHITEPAPER

Predictive maintenance based on vibrations



Introduction

This White Paper aims to discuss the benefits of connecting the Internet of things (IoT) with Machine Learning and Predictive Analysis. The Predictive Maintenance performed as a result of this, improves the way we handle our infrastructure and other important assets. A great example of this can be found in wind turbines.

Wind turbine is a device that converts the energy from the wind into electrical energy. When the turbine's blades spin, they trigger the main shaft to spin. This makes the gearbox spin as well. It is possible to monitor the state of the whole system by measuring the vibrations of the rotating mechanism.

Here, we discuss one way to monitor the state and diagnose the faults of the rotating mechanisms of wind turbines. However, it should be noted that some functionalities of the solution presented here may be useful for the diagnosis of any kind of rotating machinery faults potentially caused by vibrations.

The target audience for this document are people interested in:

- Optimizing current infrastructure/assets (e.g. wind turbine farms)
- Extending the lifetime of infrastructure/ assets (e.g. wind turbine lifespan)
- Ensuring smooth operation with predictive maintenance
- Planning maintenance budgets
- Safety

The Problem

The main problem in wind turbines are failures caused by faults in the drivetrain, led by the main gearbox.

Detecting well in time the signs that could indicate an ongoing fault in the gearbox, could:

- Reduce unscheduled downtime, i.e. improve uptime,
- Reduce or prevent altogether occurrences of catastrophic failures,
- Improve production scheduling reliability,
- Reduce maintenance costs,
- Reduce spare parts inventory,
- Reduce insurance costs (in some cases).

The question is - how to go from preventive to predictive maintenance?

Vibration analysis consideration

The gearbox system consists of a N_p-tooth pinion meshing with a N_g-tooth gear (please see the figure 1).

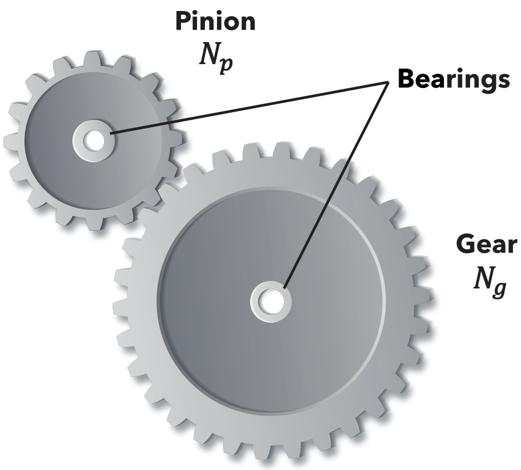


Figure 1. The gearbox consisting of the pinion and the gear

The rate of the pinion rotation is f_p and the rate of gear rotation is defined with the following formula:

$$f_g = \frac{N_p}{N_g} \times f_p$$

The gear-mesh frequency (GMF) is calculated with the following formula:

$$f_m = N_p \times f_p = N_g \times f_g$$

To measure the vibrations, different sensors are used. Placed on the specific measurement points of the gearbox, an accelerometer is a common choice. In an ideal case, just by performing the Fast Fourier Transform (FFT), it would be very easy to identify all three vibration components on the frequencies of f_p, f_g and f_m (as shown in the figure 2).

In a real-life scenario, when noise is present and some form of system defect or fault arises, it is almost impossible to detect the changes in the vibration measurements. The typical machinery faults with the gearbox system are:

- Out of balance,
- Gear misalignment,
- Gear tooth spalling,
- Rolling element bearing faults.

In order to detect changes in vibrations caused by these faults, it is necessary to perform additional signal processing techniques [1], [2], such as:

- Time Synchronous Averaging (TSA),
- Fast Fourier Transform (FFT),
- Envelope analysis in the frequency domain with amplitude demodulation, etc.

How to use these data in order to predict the behavior of the system?

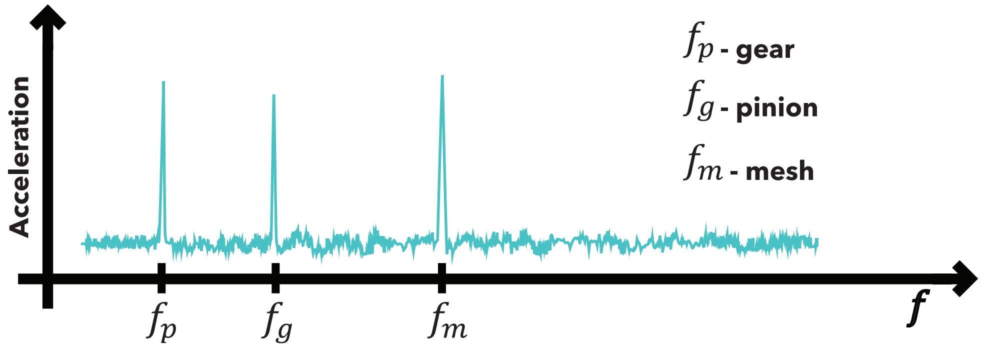


Figure 2. Vibration frequencies components

[1] Vibration Based Condition Monitoring: Industrial, Aerospace and Automotive Applications, Randall, Robert Bond, John Wiley and Sons, 2011.
[2] Practical Machinery Vibration Analysis and Predictive Maintenance, Scheffer, Cornelius and Paresh Girdhar, Elsevier, 2004.

Coming up with the solution concept

Based on the initial introduction of which vibration data to use in order to address the aforementioned problem, we can now concentrate on our solution.

To better explain the complete solution and its elements, we will use the following figure to present it.

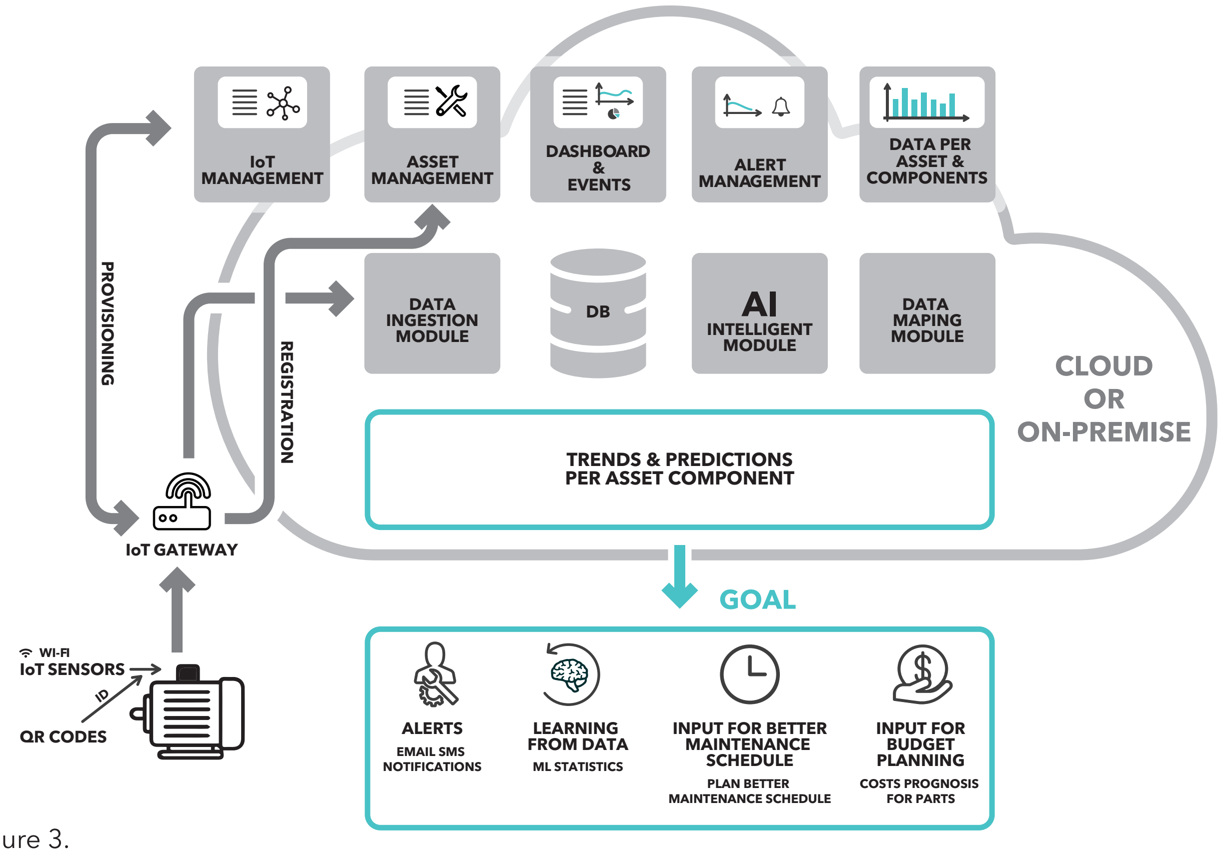


Figure 3.

SYSTEM OVERVIEW

- **Step 1: Each motor or, in this case, each asset related with vibrations, needs to be registered in the system**. This is the purpose of the asset management module, where we register an asset, the type of the asset, and its sub-components, which we mentioned earlier in the document (gear, pinion, gearbox).

- **Step 2: QR codes are used to identify the assets** in the field and to ensure that the data is mapped to the correct asset.

The QR code represents the unique ID of the asset in the system; by reading the QR code, the users are rerouted to the asset's detail page in the asset management module, where they can see all the related data and the relevant documentation.

- **Step 3: The motors are equipped with appropriate sensors**, which measure the necessary data, as described earlier in the document. These sensors are mapped with the assets in the solution, now we know to which asset the incoming data belongs when we enter it into the solution. At this stage, depending on the business needs, the sensors or a logic board can be used to analyze the data right on the spot (edge computing) and just send the relevant output data, or we can send all the data to the solution for processing. Which method will be used depends on available resources, needs and business/technical case.

- **Step 4: Provisioning the IoT infrastructure**. Security is the major concern in IoT, so every device needs to be handled in a secure way, that is why we have an IoT Management module in place. All gateways need to be registered in the solution, as does the information concerning what kind of secure protocol is used (e.g. MQTT), as well as any details about permissions, authorization, data ingestion and of any maintenance of firmware running on IoT device, etc.

- **Step 5: Gathering the data and storing it** into the solution. Here the structure splits into two branches:
  - Real time
  - Data storage

  The real time data allows us to react if some values are out of expected ranges or defined timeframes. This is connected with the alert management module, where the alerting configuration contains thresholds, special conditions, repetitiveness of events, whom-to-alert fields, etc. Alerts can be configured in different formats, e.g. email, SMS, etc.

  The data storage is used to collect over time the data for individual assets of interest. This data is used for statistical and AI analyses: to look for patterns in assets' behavior; for analyzing these patterns; remapping them to similar assets in the system and gathering insights regarding the particular types of assets – i.e. learning from data.

The key difference between our solution and other solutions in this space is that this one uses a combination of edge computing (the preliminary data analysis inside the sensoric device), stored historical data for this particular asset and data analysis based on mathematical, statistical and machine learning algorithms.

Step 6: Alerting the team that they need to respond when some anomalies are detected – the purpose of this step is to enable the maintenance team to prevent further damage. All alerts and responses are well documented in the system, which can be used for deeper analysis of the maintenance response and for insurance backup if required. If the alerts are being ignored, there is also a way to react automatically, depending on the needs (e.g. lower the RPM remotely).

MODULES OVERVIEW

The solution consists of several modules, which all contribute towards data collection, alerting and overall health of the operation.

Asset management module: besides asset registration data, it contains the history of each asset, including maintenance logs, maintenance proof documentation, manual maintenance documents, and most importantly, the overall health status of any given asset. The user is able to drill down through the asset's data structure to identify where the problem was detected (e.g. by selecting the motor entry in the asset management module, then selecting one of its flagged components, such as gear).

Dashboard & events module: the dashboard shows the overall status of the operation, including KPIs as defined. Usually a selection of KPIs is used to present the overall health status of the operation; if one of the KPIs is not as it should be, it is flagged and drill-down action can be performed to pinpoint the problem.

Alert management: used to define the rules and how events are to be handled (e.g. recurrence: if an alert is acknowledged, should it reoccur or not; if it is not eliminated, who should receive it; which communication channel is to be used per which type of event, etc.)

Data per asset & components: each asset is defined with sub-components; we can map frequency values to certain parts of an asset and identify the status/health of that asset.
Data ingestion module: responsible to get different kinds of data and remap it to a form, so that the solution can use this data.

AI-powered module: takes care of the analysis of stored data, looking for patterns in behavior which lead to bad results.

Data mapping module: data coming from edge-computing devices needs to be parsed out and mapped to specific parts of each asset in the system. E.g. data that comes to the solution aggregates multiple frequencies with the accompanying amplitudes; for each frequency, data mapping is executed to be able to attach it to a particular component of the asset (e.g. gear, pinion). This input is then used for alerting as needed.

Trends & predictions: this module takes the outputs of the AI-powered module and the outputs of the mapping module and applies these patterns/results to all the assets of the same type in the solution. On top of this, statistical/mathematical functions are applied to determine trends in the assets' behavior.

The solution brings its users the following advantages:

Alerts: the maintenance team gets alerts when some unexpected behavior is detected or there is a high probability of an issue arising. Using smartphones, the team members can get SMS prompts in real-time to alert them where they should focus their attention. The goal is to prevent further damage and offer suggestions as to how to prioritize tasks, allowing experts to judge by themselves which actions to do first.

Learning from data: the idea is to collect and analyze the data to identify the healthy operation behavior and set it as the expected operating conditions. Learning through time, we can expect different kinds of behaviors (on weekdays, during certain hours in the day etc.). The other approach is looking for patterns in the data and applying them to other assets that fit this environment and especially learning from the historical data when a failure occurs.

Optimizing downtime due to maintenance: downtime can be costly, especially if it is unscheduled. The idea is to combine data insights with predictions to help with scheduling maintenance in such a way that all assets which are approaching to a varying degree of their preventive maintenance window, get combined in the same scheduled downtime. Although an asset could still work fine for a short amount of time, it is better to perform its premature maintenance, than risk it undergoing an unscheduled failure, since such sudden downtime is much more costly than the preventive approach. Prevention scheduled on the basis of data insights results in a better maintenance and saves costs over the long term.

Help with cost planning: because the solution can foresee failures by inferring them from experience, statistics and predictions, it can predict costs well ahead of time. Thus the data insights can help with planning the maintenance budget.

Prototyping / PoC

Let's explore how we can quickly prototype such a solution:

1. For cloud or on-premise services, we would use the Forensixx solution, which includes:
   a. Asset management
   b. Dashboard & events
   c. Alert management
   d. Asset details
   e. Data intelligence modules

2. For IoT infrastructure provisioning, we would use an open-source solution which includes:
   a. Gateway management
   b. Sensoric management

3. For creating and manipulating predictive models, we would use:
   a. KNIME open-source solution
   b. Or any other, e.g. Azure ML, Tensorfow, etc.

4. For integrating all these parts, including legacy systems necessary to gather the data or serve other business needs, we would develop a tailor-made integration module.

EQUIPPING MOTORS WITH IOT

Enabling the IoT option on existing and new motors requires two main features:

- measuring and tracking, and
- connectivity.

By using various sensors on the equipment, among them the vibration sensor as the principal one, we can measure and track different signals.

The connectivity enables us to integrate the data collected by the sensors into data ingestion.

POWERING THE SENSORS

Taking the above into account, we can conclude that sensor-integrated solutions for fast prototyping are the perfect choice when implementing a quick solution. For example, SensorTag from Texas Instruments provides an integrated solution with all the necessary sensors and BLE/WiFi connectivity for data collection. An additional onboard Cortex-M microcontroller is capable of performing basic DSP tasks on measured signals.

These solutions are optimized for low-power applications, so SensorTag can be battery powered and have years of battery lifetime from a single coin cell battery. This is also a perfect choice when the retrofitting requirement is of high importance.

When it comes to industry-grade solutions, one good example is the SSA100 vibration sensor, with IP67 protection and EMC compliance. The CAN communication interface ensures it can be easily paired with the gateway.

CONNECTING THE SENSORS TO GATEWAYS/ CLOUD/ON-PREMISE

The next step is to integrate the sensor data. Regardless of whether it is needed on-premise or in the cloud, the use of an industrial-grade gateway is the optimal solution. A proven option is the multi-service IoT gateway ReliaGATE 10-11, from Eurotech.

This is a cloud certified industrial-grade gateway, IoT-ready and covering a wide choice of connectivity options with sensors and cloud services. It has been designed to comply with global certification requirements for industrial and lightly rugged applications.

By using various I/O interfaces from the gateway, it is possible to expand the sensor network with other types of sensors and measurements.

INTERFACE

The interface from the gateway serves to collect the data from the sensors. Using ReliaGATE 10-11 gateway, it is possible to connect it using open-source Eclipse Kapua – a modular IoT cloud platform to manage and integrate devices and their data (https://www.eclipse.org/kapua/). Using the Kapua's REST API, the data can be gathered and sent to Forensixx.

Another option is to send the data directly from the gateway to Forensixx, using MQTT.

FIRMWARE (EDGE-COMPUTING)

The chosen gateway ReliaGATE 10-11 is perfectly capable of performing a wide range of edge-computing tasks, especially for DSP algorithms on the signals from the sensors. The gateway is run by embedded Linux, and main firmware for edge-computing would be designed in C++. Using highly optimized, proven open-source processing libraries from ARM for Cortex-A microprocessors (https://github.com/ARM-software/CMSIS_5/tree/develop/CMSIS/DSP) would ease the firmware design and center the main focus only on tasks of interest.
After the necessary signal processing tasks, the gateway has all the required data to monitor the system state by detecting any changes that occur over time. This is the basis for the gateway to send:

- deltas,
- events,
- alerts, and
- other information of interest

to the data ingestion module, using TCP/IP over the 3G GSM or Ethernet connection.

BLOCK SCHEME

The following figure shows a block scheme of the motor, equipped with sensors connected to the gateway.

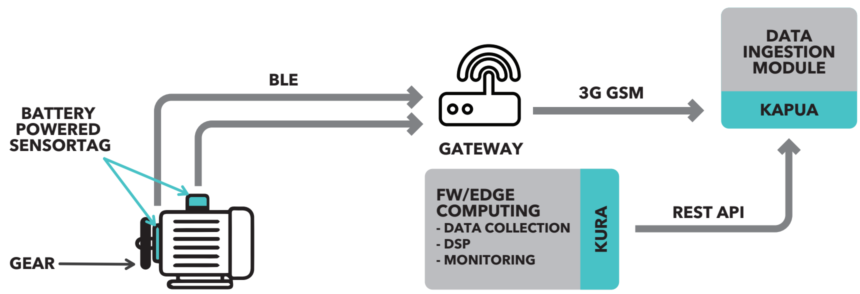


Figure 4.

Using AI to enhance the solution

In the first phase, when there is no data available from the field, statistical and mathematical functions need to be applied, together with threshold logic. Over time, when the data comes in, we can apply machine learning approach to it.

WHAT WE ARE TRYING TO SOLVE WITH AI

The major goal of an AI-powered module is to predict failures, based on data patterns gathered over time, and generate the following reports:

- How many assets may have a problem in the following time periods (1m, 6m, 12m)
- What kind of issues we might expect (gear, pinion, misalignment etc.)

Based on the vibration analysis we mentioned earlier in the document, the following data is taken into account:

- On the IoT side:
  - Temperature of the motor
  - Vibrations on different segments of the spectrum
  - RPM of the motor
  - The telemetry data recorded over time
- On the asset management side:
  - Designated RPM of the motor
  - Type
  - Age
  - Location
  - Estimated heathy running conditions
  - Components of the asset (gear, pinion, etc.)

THE CONCEPT OF AN AI-POWERED SOLUTION

The following figure represents the concept of a solution using AI. The telemetry data of the asset, per particular asset's component, is gathered over time. When a failure occurs, the telemetry data from the past is used to examine the history of the asset in question and learn from it.

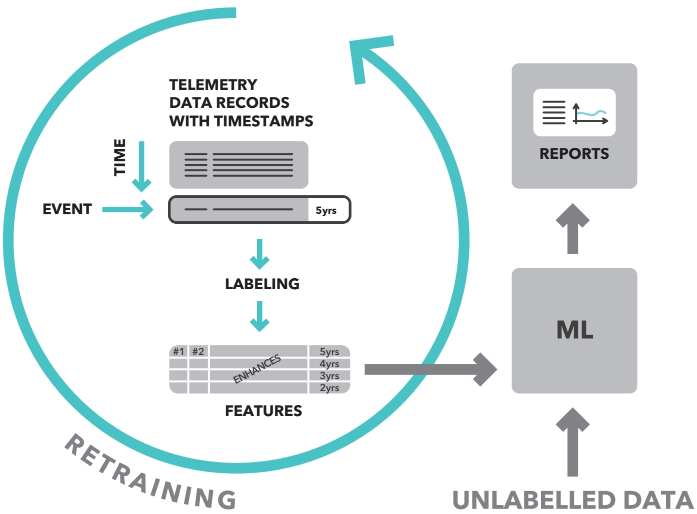


Figure 5.

The figure shows that we are collecting the telemetry data over time; when an event (a problem) occurs, we review historical data and label it. In this way, we can use it for machine learning, to discover patterns which can be applied to other similar assets working in similar conditions. The goal is to deduce the remaining lifespan on other assets.

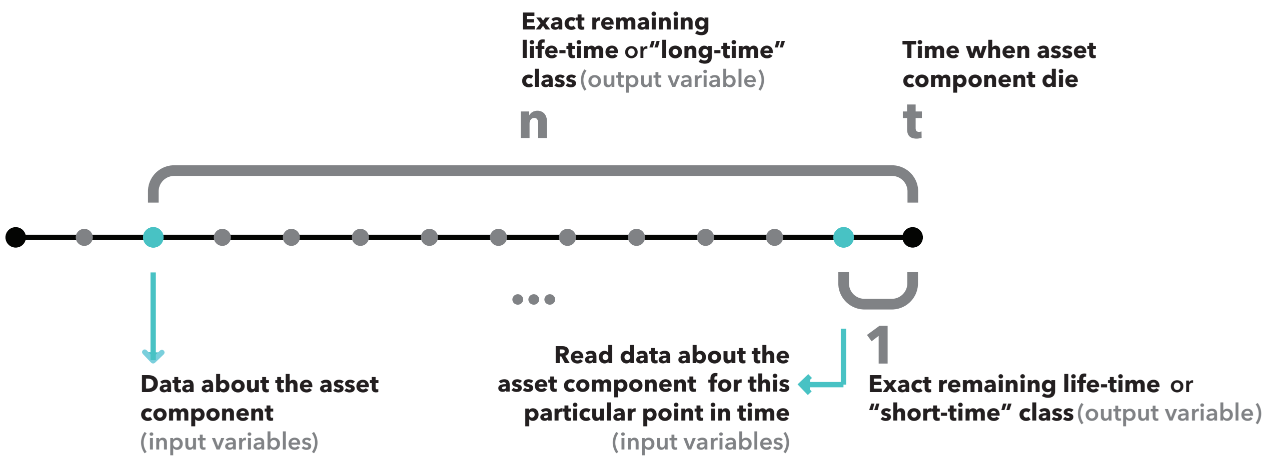


Figure 6. Gathering the data for supervised machine learning process

DEALING WITH THE DATA

Usually, the data comes from various sources; some are new and some come from legacy systems. In practice, this means that data manipulation needs to be introduced; we need rules about what to do with any missing fields, or in using the same units (e.g. mA, A), etc. All these issues need to be addressed before the data can be used by the solution.

ENHANCING THE DATA WITH FEATURES

Based on the concept we defined, here are some additional features, which can help with the Machine learning approach:

- Average/Min/Max value of temperature to date
- Average/Min/Max value of vibration amplitude per frequency to date
- Expected RPM (typical working load)
- Average/Min/Max value of RPM to date
- Asset type

The combined data is the starting point for creating a model; in time, we can introduce or remove features (and the corresponding data) to update and/or revise the predictive models from time to time.

In our case, we are using a supervised Machine Learning approach, where models are trained on a collection of labelled examples. The dataset would consist of asset instances we describe using the data features listed above (input variables) and their labels (output variables), e.g. known lifespan remaining. The next figure shows an example of how we can represent the data in a tabular way.

Input variables					Output variable
Avgerage temperature	RPM deviation	Location	Average vibration X1 amplitude	...	Remaining lifetime
...	...	...	...	...	...
...	...	...	...	...	...
...	...	...	...	...	...
...	...	...	...	...	...
...	...	...	...	...	...

GENERAL APPROACH TO DATA ANALYSIS

The acquired data is essential in the process of model training and evaluation. We first analyze it in order to understand it and to gain new knowledge. In addition, we can apply various filtering, cleaning, transforming and feature engineering techniques to the data.

MODEL TRAINING AND EVALUATION

This phase includes a number of sub-steps, such as candidate algorithm identification, valuation method(s) and measure(s) selection, model training and validation, model selection, model testing, etc. The goal is to find the best Machine Learning configuration for the problem at hand and to estimate its future performance.

If the outcome of the Machine Learning model belongs to a finite number of categories, the process is called classification. If the outcome is a continuous quantity, the process is referred to as regression. In the case of predicting lifetime of our assets, both approaches can be applied:

1. Classifying assets into a fixed number of classes, such as short, medium and long lifespan (classification task) or
2. Estimating the exact number of remaining time units (hours, days, months, etc.) for the assets (regression task).

The first option is useful if one is not interested so much in detailed prediction, but rather in high-level categories. The second provides more specific predictions, but it is challenging to predict the exact lifespan of an asset, especially as there are not many labelled training instances available.

Various algorithms could be applied for estimating the remaining lifespan of the assets, which depend on real business cases, for example: K-Nearest Neighbors, Support Vector Machines, Neural Networks, etc.

Several candidate algorithms are chosen and applied to a given dataset in order to find the best performing one for the particular business case.

We split the dataset into a training set and a test set. Performing cross-validation on the training part, by changing various parameters, helps us choose the best model. Testing the chosen model on the test set then helps us estimate whether its performance is unbiased in general.

We assess the performance by calculating evaluation measures, such as accuracy, precision and recall for classification and mean absolute error (MAE), root means square error (RMSE), etc. for regression. Lastly, using the best performing configuration, the final model is trained on all the available labelled data.

APPLYING THE MACHINE LEARNING MODEL ON NEW DATA

After training the Machine Learning model, we use the model to estimate the remaining lifespan of assets which are still functional. The individual results can be aggregated in order to produce a summarized graph (see the following figure), which shows the estimated remaining lifespan of assets, per class or time unit.

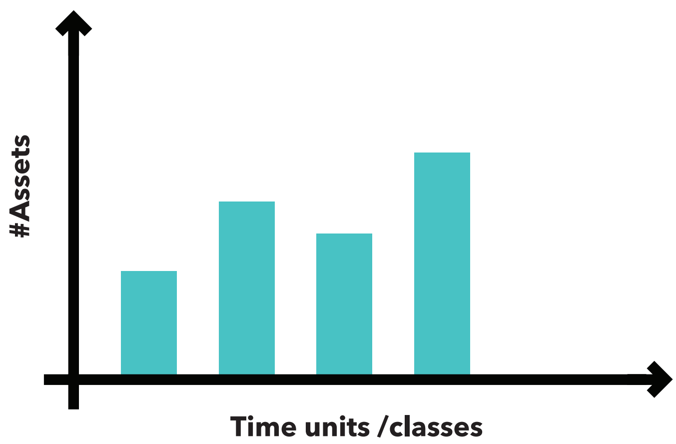


Figure 7. Summarised graph for asset lifespan estimation

USING THE PREDICTION MODEL

Models created for specific business cases are propagated as Web Services, which makes the solution architecture loosely coupled, so that it can be integrated into any kind of solution.

The principle is that the assets taken from the Asset Module are periodically checked against the prediction model, for reporting/planning needs.

IMPROVING THE MACHINE LEARNING MODEL OVER TIME

The solution doesn't end here; the model needs to be periodically retrained with new data. Some additional optimization of the data and algorithm tuning may also be required.

The model can be improved over time by adding new examples to the training dataset and periodically retraining the model. By applying such an approach, the model constantly gains new knowledge and is updated with the latest trends and changes.

The model retraining can be implemented in different ways; for example:

- by applying a strategy which enlarges the training dataset, by continuously adding new labelled examples, or
- by implementing a mechanism which forgets the oldest instances, while the new ones are added to the training dataset, etc.

Also, we need to deliberately let some percentage of the assets run into trouble, to check if the prediction model really works.

Wrap-up

In our concept, we used a step-by-step approach in order to solve the business case:

- we checked what issues contribute to the problem
- we found out how to deal with these issues
- we identified what is needed to find an adequate solution
- we created a Proof-of-Concept solution with minimal effort and cost
- we proved that the solution has true benefits towards attaining our goal.

In light of the information above, we consider that the AI-assisted vibration monitoring can bring value to wind turbine businesses.

Comtrade d.o.o.
Letaliska cesta 29b
1000 Ljubljana, Slovenia
T: +49 711 22254 233
E: info.de@comtrade.com
www.comtradedigital.com