

Prognostics and Availability for Industrial Equipment Using High Performance Computing (HPC) and AI Technology.

Peter Darveau, P.Eng., MBA^a Meeta Kalra, M. Sc.^b

^aDirector, Engineering - Hexagon Technology Inc, Oakville, Ontario, Canada

^bData Architect - Dataperformers, Montreal, Quebec, Canada

Abstract. The Industrial Internet of things (IIoT) enabled smart system has entered into a golden era of rapid technology growth. IIoT is a concept to make every system interrelated such that they are able to collect and transfer data over a wireless network without human intervention. In this paper, we discuss the development of an IoT enabled system to monitor the vibration signature of equipment as part of prognosis and availability management system (P&AM) that serves to prevent unplanned operation downtime and catastrophic failure of a whole system. In order to simply the complexity of processing video content and performing inference, the Intel OpenVINO platform was selected because of its simplicity, portability across Intel AI processors, performance and comprehensiveness of its analytical and diagnostics capabilities that can be tested in Intel's DevCloud. The IIoT system consists of a High Performance Computing (HPC) platform based on Intel's Xeon processors and Movidius AI accelerator, Intel's OpenVINO toolkit for AI, a Regul high performance programmable controller capturing vibration data through sensors and a low-latency network connection. Notifications of anomalies are sent to a smartphone. This paper reveals an approach for the features extraction and selection, known as feature engineering, of the equipment component we want to protect. Feature engineering is the first step for the P&AM of these components and extends to the whole system. The broader aim of this paper is to help technical leaders at the exploring or experimenting stages of their AI framework to learn the concepts of implementing algorithms using datasets that have real value to their companies. Datasets generated and referred to in this paper were generated by simulation under various material failure scenarios.

Keywords: Feature engineering, vibration, high performance computing (HPC), dataset, prognostics and availability management (P&AM)

1 Introduction

OEE (Overall Equipment Effectiveness) is the gold standard for measuring manufacturing productivity. It identifies the percentage of manufacturing time that is truly productive. An OEE score of 100% means you are manufacturing only good parts, as fast as possible, with no stop time. In the language of OEE that means 100% Quality (only Good Parts), 100% Performance (as fast as possible), and 100% Availability (no Stop Time). The biggest problem with OEE is that it doesn't build risk of disruption into the equation. Prognostics and Availability Management (P&AM) is an engineering discipline that identifies fault severity and predicts the remaining useful life (RUL) of the target system. P&AM is the enabling technology towards condition-based maintenance (CBM), which is the future maintenance strategy as opposed to

corrective maintenance (CM) and periodic maintenance (PM). Numerous books have been published recently addressing various aspects such as signal processing, data driven diagnostics, prognostics, and practical applications. In general, P&AM consists of three steps: (

- 1) Data acquisition and features extraction
- 2) Fault diagnosis
- 3) Failure prognosis.

In machinery, gears, bearings, impellers, motors, shafts, footings, foundations are all important components because they transmit or absorb power while supporting applied loads in the system. Therefore, their unexpected failure and degradation during operation can lead to economic loss and catastrophic accidents. In fact, the failure of these components accounts for a large proportion of whole system failures, which is the main reason for focusing on P&AM [1] [2]. Thus, the feature engineering of these components is of crucial importance. While there have been many studies in this area, their concern was mostly on new algorithm development including the deep learning (DL) approaches recently gaining popularity. Relatively little attention has been given to beginners to implement value-added DL applications. In light of this shortcoming, this paper also represents a roadmap for performing feature engineering.

As P&AM is performed from signals that are obtained from sensors such as vibration, acoustic emissions, acceleration or other signal transmitters, it is crucial to remove undesired noise and extract the valuable information called features from the raw signals in the first step. Based on the extracted features, the fault mode and its severity are identified through data from simulation for various material failures, and RUL is predicted using the prognosis algorithm. While there are many useful features for this purpose today, good features vary depending on P&AM steps and applications. For example, in the diagnosis, features should show a clear difference between the normal and fault states to distinguish the health condition. In the prognosis, on the other hand, features showing a monotonic trend over time are considered good indicators for RUL prediction. In this context, the process of extracting relevant features is also called “feature engineering”.

Advances in high performance computing (HPC) are unlocking ground-breaking possibilities for a whole range of industries. Yet, it is not just computing power gains that companies stand to benefit from. As new technologies, such as Artificial Intelligence (AI) and the Internet of Things (IoT), continue to evolve, organizations are presented with more opportunities to apply HPC capabilities in new ways. In this paper, we present a system architecture to take real-time streamed sensor information and process it to perform P&AM. Using Intel high performance chip technology and OpenVINO toolkit for IoT, the process of analyzing data is simplified by using deep learned models to identify patterns on the sensor signal rather than more intensive continuous computing on the discrete raw data. Therefore, we perform analysis in the time domain instead of the frequency domain. This allows us to perform an optimal P&AM analysis regardless of the feature(s) deemed critical to monitor for the application at hand. The data coming from appropriate sensors in locations determined by the results of feature engineering is captured by a programmable controller (PLC) which in turn sends the signal to a centralized Xeon CPU-based server which combines inputs from all sources and streams the signal in real-

time to the Cloud to perform signal pattern recognition and detection and sends resulting information to an app on a smartphone.

2 P&AM Framework and Datasets

The P&AM framework can be divided into several steps as shown in Figure 1. After the data acquisition, the signal processing step is performed which aims to remove noise from the raw signal to obtain only the necessary information for the diagnosis and prognosis. Discrete signal processing is performed to obtain only the necessary information for the diagnosis and prognosis. Residual fault signals are then enhanced by signal enhancement techniques. If necessary, the signals, whether in their raw form or after processing, can also be decomposed into various components to facilitate the exploration of fault information. Detailed information about the techniques is addressed in a previous study [3]. Once the signal processing is done, the feature engineering process is conducted next, where the features that represent the fault or degradation of the system can be extracted, evaluated, classified and selected as shown in Figure 1. Finally, by exploiting these features, fault diagnosis is performed to classify the state of the target system or failure prognosis is performed to predict when the failure will occur in the future.

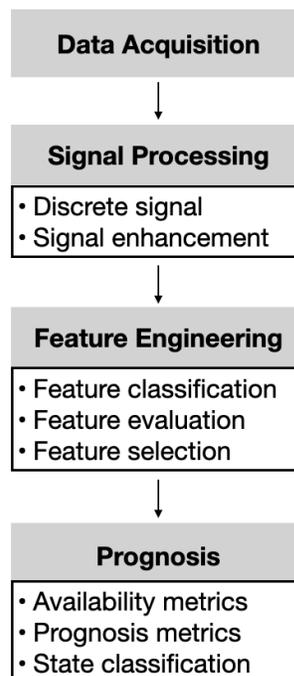


Figure 1 - Framework Prognosis & Availability Management (PAM)

Note that feature engineering requires substantial skill and expertise in data science, which is the key to the success of the subsequent steps: diagnosis and prognosis. The feature engineering process is illustrated in Figure 2.

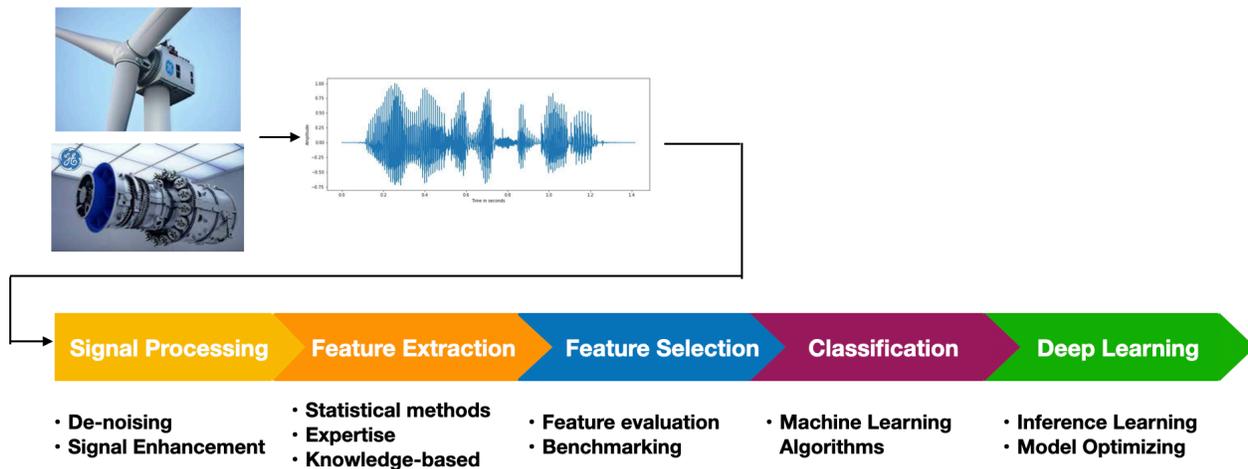


Figure 2 - Data-based feature engineering procedure

In P&AM study of equipment and machinery, the general method of data acquisition is to measure vibration by attaching an accelerometer to the critical components. Gathering the data is, however, too expensive, requiring a test rig, sensors, and time-consuming tests. To avoid this, environmental vibration analysis can be obtained for critical components under different material failure models through simulation or a digital twin, a virtual representation that serves as the real-time digital counterpart of a physical object. In essence, there are two categories of data that are needed for P&AM. Diagnostic data are obtained by running the component under normal and fault conditions, respectively. Prognostic data are obtained by running the components constantly from the normal to the failure state. The desired conditions for diagnostic and prognostic data can be chosen under very specific mechanical conditions and loads chosen by the design and mechanical engineers. Heat maps are used to assess the condition under which we want to generate the data as shown in Figure 3.

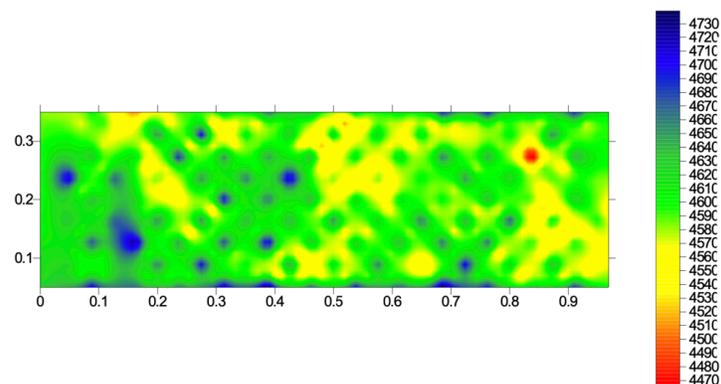


Figure 3 - Heat map for material diagnostics and prognostics analysis

Datasets derived through simulation or a digital twin provide two main advantages: provide high-value data specific to a business' assets and, a way to improve Return of Investment (ROI) when deciding on IoT and ML projects because of the focus on risk control to critical business assets disruption rather than looking to expose it before its trips the company. Once the datasets are created, feature engineering can be performed to determined the best way to create the machine learning model to best perform P&AM.

3 Feature Engineering

It is important at this stage to mention why we need to perform feature engineering for P&AM. Traditionally, machine learning relies on a prescribed set of “features” that are considered important within the dataset. In our P&AM application, features relevant to a machine's likelihood to fail might be the vibration signature of a bearing, a gear, and structure [4]. The process of building features into a machine learning algorithm is known as “feature engineering.” Feature engineering requires deep expertise in a given subject (here, the specific machine) and can be a very labor-intensive process for the data scientist.

Deep learning is a type of machine learning that has received increasing focus in the last several years. With deep learning, the algorithm doesn't need to be told about the important features. Instead, it is able to discover features from data on its own using a “neural network.” The name is inspired by a mathematical object called an artificial neuron that “fires” if the combination of inputs exceeds some threshold, just like a neuron in the brain does. Artificial neurons can be arranged in layers, and deep learning involves a “deep” neural network (DNN) that has many layers of artificial neurons.

Deep learning enables us to avoid feature engineering altogether. Given enough “labeled data” (i.e., images of known vibration signatures or patterns) and the right tuning, a deep learning model will identify the most relevant features from the data on its own. Deep learning represents a conceptual shift in thinking, from "How do you engineer the best features?" to "How do you guide the model toward discovering the best features?" Due to the specialized and custom nature of the system we want to implement, it is the latter question that we answer in the following sections.

4 Feature Extraction

Feature engineering begins with feature extraction from data in its raw form or after going through noise removal by signal processing depending whether the data is acquired in a live or simulated setting. The features are usually divided into three categories: time domain, frequency domain, and time-frequency domain. Among these, time-domain features are the easiest to analyze and most widely used, especially with industrial control systems that capture data and perform functions based on the time domain [5]. These are addressed in the following sections.

In feature engineering, several signal processing steps are usually taken and appropriate features are extracted from each step that enable fault identification. This is shown in Figure 4, which indicates that the raw data are processed step by steps and after each step, relevant features are extracted. An example of the signal processing process in time domain and its Fourier transform in the frequency domain are also illustrated. Note that the features are divided into two groups:

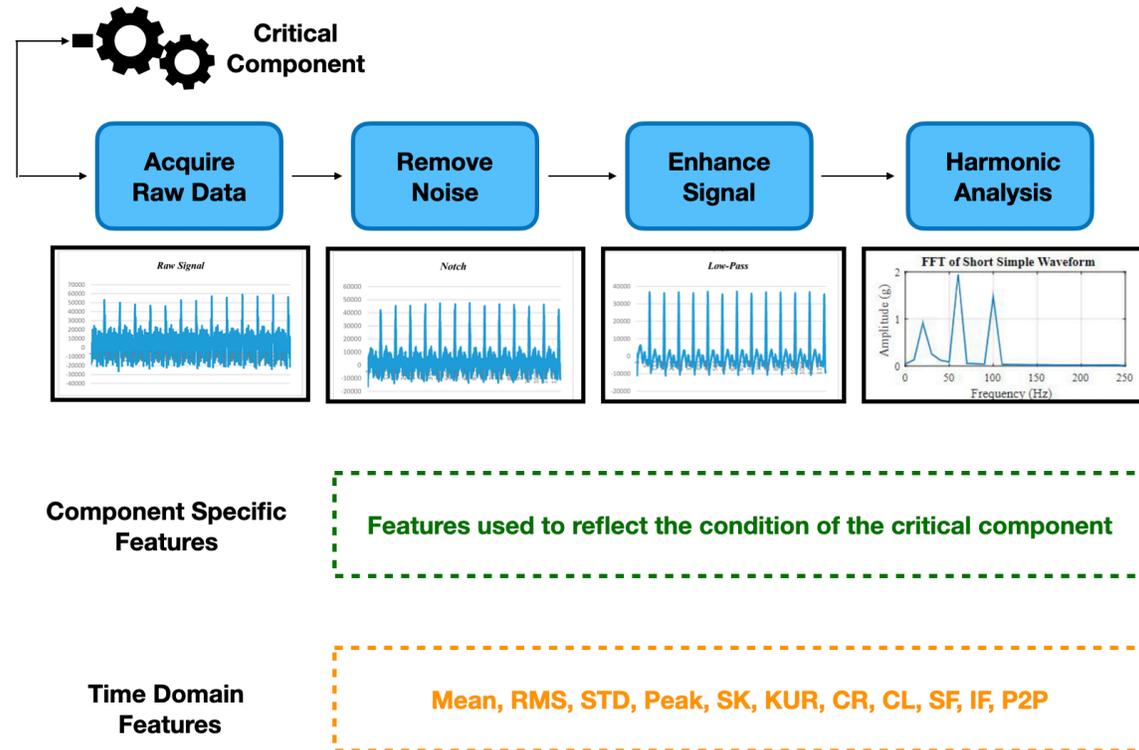


Figure 4 - Feature extraction process

Component specific features shown by the green dotted box, and time-domain features in the orange dotted box. Note that the component specific features have generally been developed for those components for fault diagnosis and can be found in many journals and articles to that affect. Also, more and more component or Original Equipment Manufacturers have this important data important for understanding key maintainability factors of their products.

While there are many benefits to performing harmonic analysis and generating FFT on time-domain data, it may not be necessary to add this compute intensive functionality to get the most out of a P&AM system [5]. Recall that the time domain features can be extracted either from the raw signal directly or after taking this step depending on the problem. The engineer and data scientist will want to do more investigative work to determine the need to work in the frequency domain or if a simpler system based on time-domain analysis will accomplish the task. A list of general time domain features are listed in Table 1.

Abbreviation	Full Name	Brief Explanation	Formula
MEAN	Mean	Average	$\frac{\sum X}{N}$
RMS	Root mean square	Value that generally tends to get bigger as the degree of fault in the bearing increases	$\sqrt{\frac{\sum X^2}{N}}$
STD	Standard deviation	Value representing the dispersion of a signal	$\sqrt{\frac{\sum (X-\bar{X})^2}{N-1}}$
PEAK	Peak	Maximum value of signal absolute value	$\max(X)$
SK	Skewness	The asymmetry of the probability density function of the vibration signal	$\frac{\frac{1}{N} \sum (X-\bar{X})^3}{STD^3}$
KUR	Kurtosis	The sharpness of the probability distribution of the vibration signal, and if this value is close to 3, it is closer to normal distribution	$\frac{\frac{1}{N} \sum (X-\bar{X})^4}{STD^4}$
CF	Cres factor	The ratio of peak values to the RMS of a signal	$\frac{PEAK}{RMS}$
CL	Clearance factor	Peak value divided by the square of root mean	$\frac{\max(X)}{(\frac{\sum \sqrt{X}}{N})^2}$
SF	Shape factor	RMS divided by mean	$\frac{RMS}{MEAN}$
IF	Impulse factor	The ratio of peak values to the mean of a signal	$\frac{PEAK}{MEAN}$
P2P	Peak to peak	The difference between maximum and minimum values of the signal	$\max(X) - \min(X)$

Table 1 - General Time Domain Features

Figure 5 is the result of extracted time domain features for an example dataset: 6 normal of a critical component in a machine and 11 fault data, in which the acronyms are found in Table 1. Note that each set of feature data is normalized by mean and standard deviation, respectively. In the results, most of the features classify the fault (x) from the normal (o) well, but some such as SK and SF do not, which means that further processing is necessary to select only the useful features.

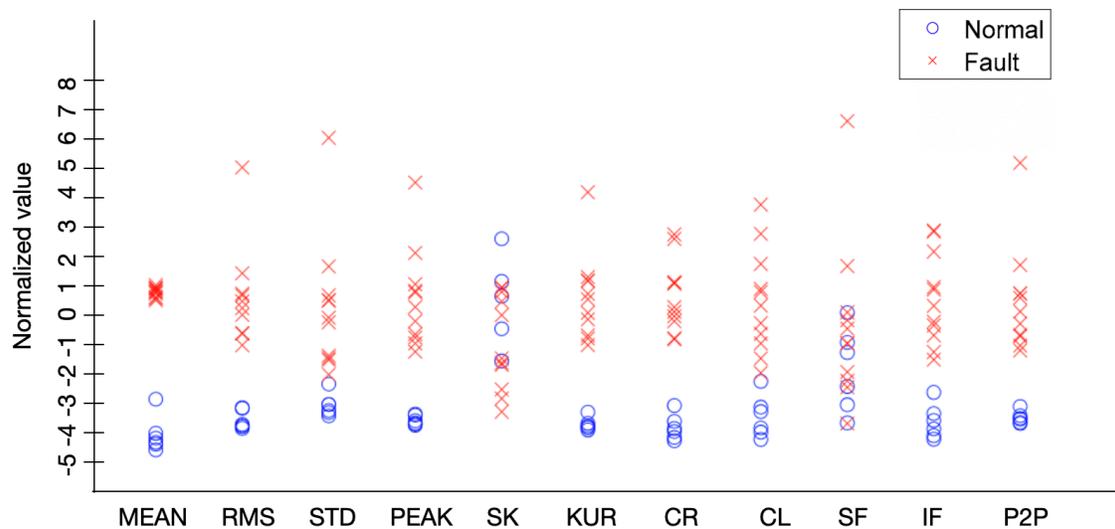


Figure 5 - Normalized time domain features of a critical component

5 Feature Evaluation and Selection

Feature evaluation is to assess how well the features can classify between the normal and faults conditions. Once a method is done, a few of the most significant features can be selected for the efficient fault classification. Feature evaluation involves analyzing the class separability per certain criteria. Here, we need to look at the number of classes represented in our datasets. Referring back to Figure 5, we note that there are two classes (normal and fault) that distinguish our datasets. However, if the critical component is made up of more than one unique moving part like a gearbox, a solenoid valve or a compressor which can exhibit failure modes based on different sub-component failures there can be three or more classes. Figure 6 shows the normalized time domain features for an sample component with three classes.

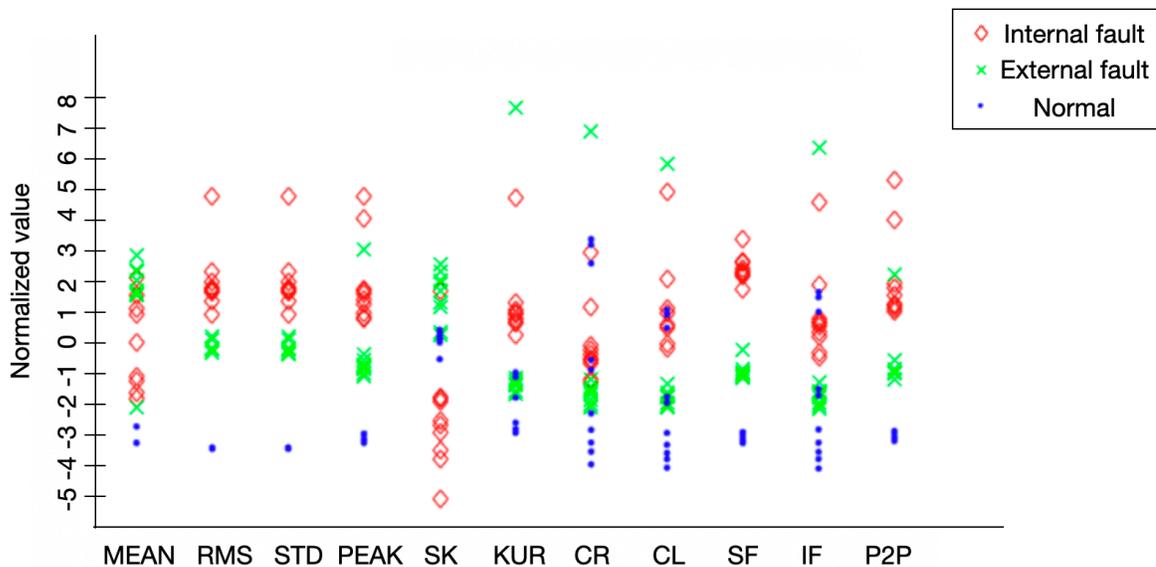
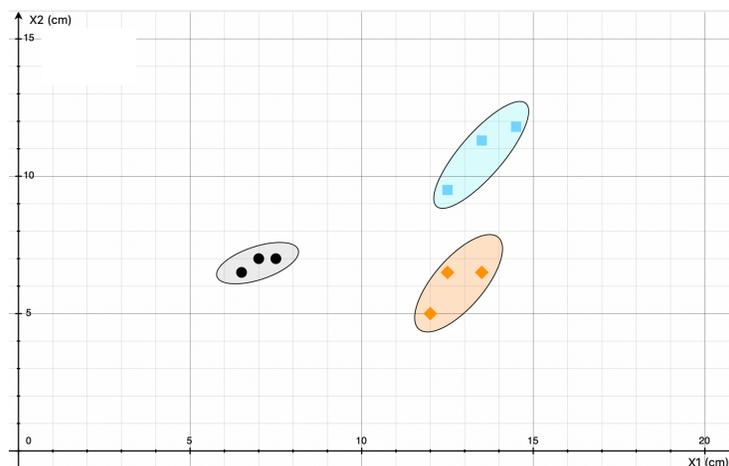


Figure 6 - Normalized time domain features of a critical component - 3 classes

Feature selection will occur in two major stages: Diagnosis and Prognosis. The diagnosis stage will reveal the features that are the most likely to identify the presence of a problem by examination of the vibrations in the time domain. Prognosis reveals the feature that is the most likely to tell us about the risk of a failure and hence gives us information about the risk to the availability of the equipment. We examine these two stages below:

5.1 Diagnosis

Generally, to determine the best features to select, we look to determine the samples that cluster tightly around their class means and seek the situations where classes are well separated from each other. To do so, performing Fisher's Discriminant Ratio (FDR) and looking at the Probability Density Functions (PDF) of each time domain feature for each class will reveal the feature that is most clearly distinguished from the others in a range. This works well for datasets with two classes. For datasets with three or more classes, the J3 function (expanded FDR) can be used to measure the separability of the features between classes. In both cases, we are looking for the scatter plot of the pair of features with a high FDR or J3 as shown in the example plot below. It is left to the reader to look up the mathematical functions of these two classification separability analysis methods. The picture below is a good example of a feature pair being a strong candidate for selection.



5.2 Prognosis

In the final step of our feature engineering process, we look at the prognosis which is the forecasting of the likelihood of a component failure. In other words, which feature(s) are the most heavily weighted. Because industrial environments have a tendency to be noisy, it is best to use an index that decomposes sub-bands rather than the raw signal to capture

the most information. The latest scientific literature, proposes the following criteria for feature selection in such an environment and its based on the calculation of average of the three following metrics for each feature:

- $\text{Corr}(T, X)$ - Correlation of time and feature which is a measure of linearity
- $\text{Mon}(T, X)$ - Monotonicity of time and feature which is a measure of the degree of continuous increase and decrease of the feature over time
- $\text{Rob}(X)$ - Robustness which has to do with the noise arising from the raw signal.

This index gives us the feature with its likelihood of predicting unavailability. A higher index for a feature thus tells us that it is a good candidate to select for a monitoring system to be effective at P&AM using our AI system as it can be monitored in the time domain. Figure 7 shows the results of the calculations for a given application. In this example, P2P (difference between maximum and minimum vales of the signal at a given time) shows the highest value. This gives a strong indication that the feature P2P is the best feature for P&AM.

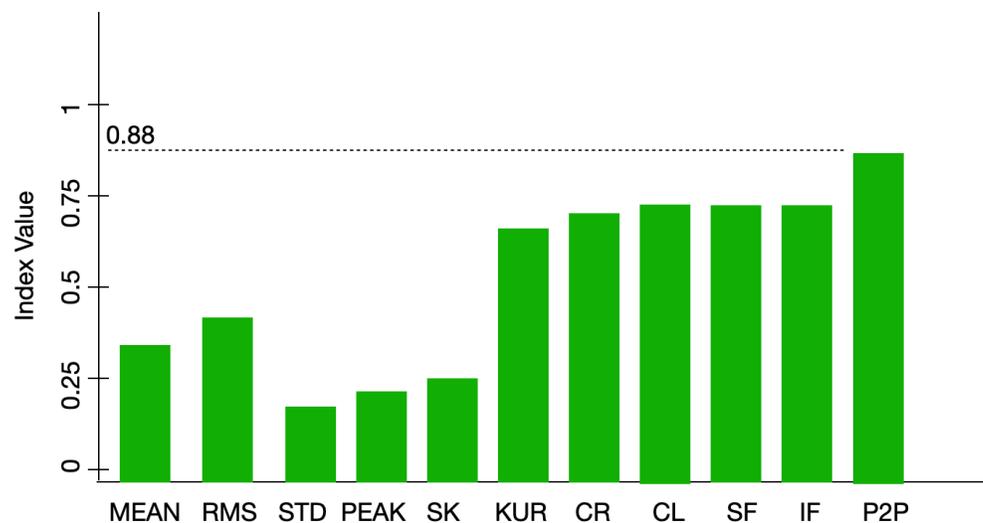
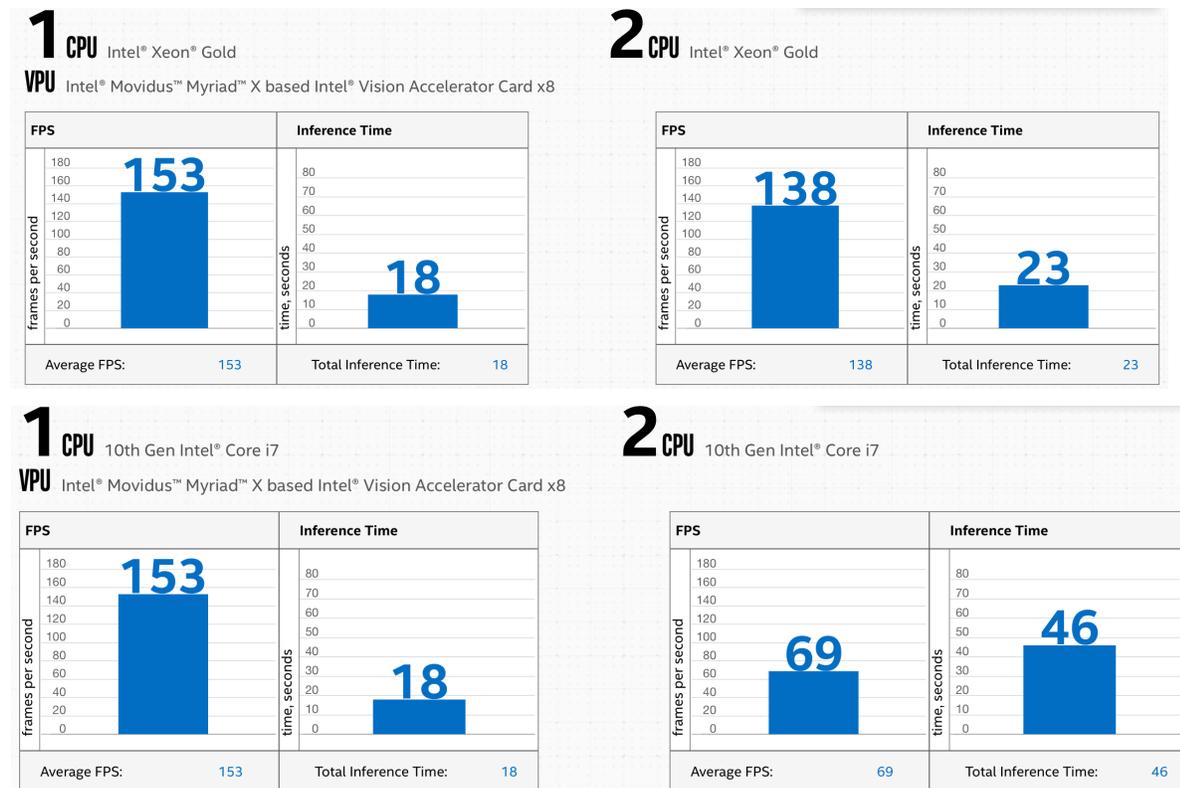


Figure 7 - Feature selection criteria

6 Proposed System Architecture & performance

The P&AM system is a system based on Industrial Internet of Things (IIoT) with the capability to notify a user when an anomaly is detected. The system is implemented within three levels of the ISA-95 technology stack standard. The Level 1 implementation involves a Regul PLC to capture the raw data from the vibration sensors because of its robustness to the industrial environment where the equipment is located. Since vibration data is provided by a transmitter that de-noises and conditions the signal, the raw data can be used directly to perform frame

prediction on the streamed signal/time trend. The Level 2 implementation involves a server to process the raw signals and stream them in the time domain to the L3 Cloud server that performs the video inferencing algorithm with a combination Intel Xeon or Core i7 CPU and Movidius VPU AI accelerator for added performance. From benchmarking tests on Intel DevCloud, we determined that adding the AI accelerator for the Level 3 implementation, which is the most compute intensive, was the most effective at providing optimal performance to reach 153 frames per second as shown below, which ensures that the streaming is smooth.



The proposed system architecture in Figure 8 works as follows. As we determined above, it became clear through our feature selection that we should monitor P2P for an optimal performing P&AM system. The Level 3 system's AI model was developed using examples of snapshots from simulation of the vibration patterns or signatures of the components being monitored. As the raw signal is being streamed, the AI inference engine continually checks the video stream for the P2P patterns used to teach the model. When no pattern is matched, the system captures the unknown vibration signature and sends an anomaly warning to a user via an app on a smartphone.

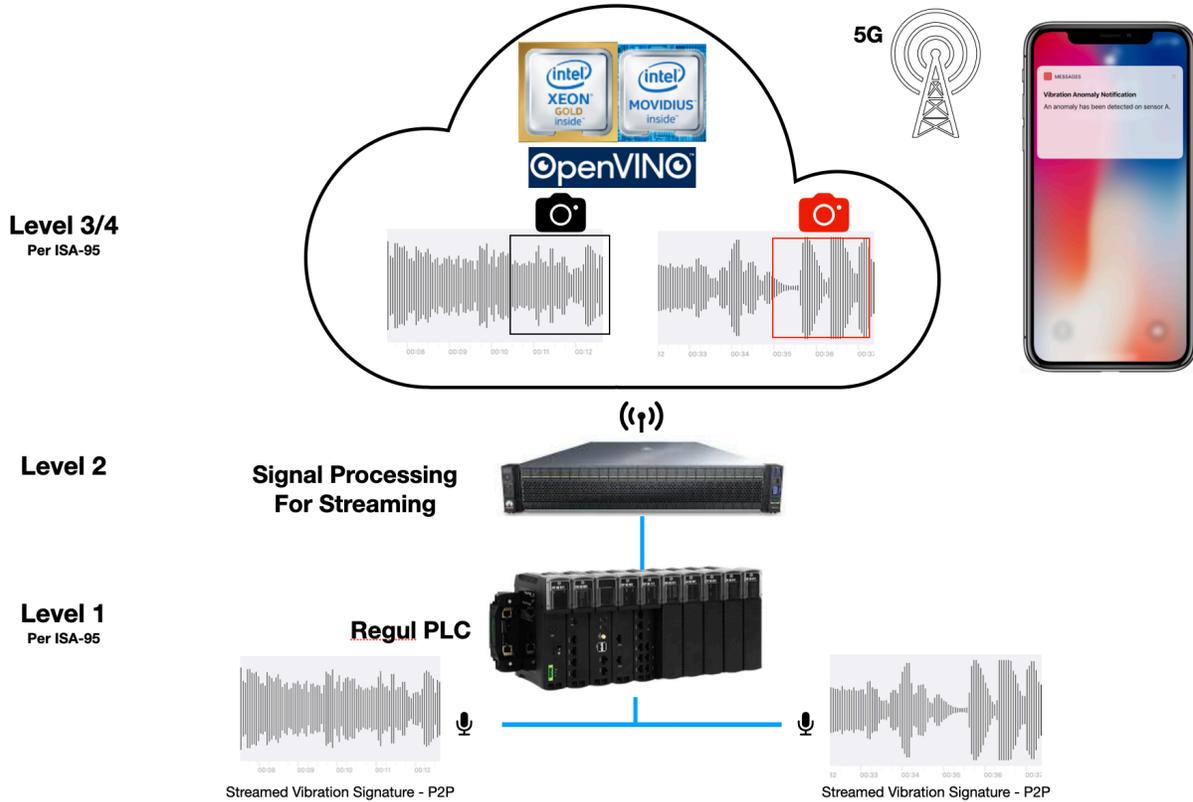


Figure 8 - System architecture

An actual set up consists of extruded-slotted aluminum framing designed in an ergonomically construction as shown in Figure 9. Highly modular and connectorized, the system is designed for quick maintenance and parts replacement using IEC 61131, ISO 9001 standards as well as and open software standards for easier compliance and upgrades with evolving technologies.



Figure 9 - As-Built diagram and Pilot system

7 System Software

The P&AM system is implemented on Intel Xeon CPU-based platform in Linux Ubuntu OS environment which supports TCP/IP, HTTPs, JSON, OneAPI, SSH. Intel's OpenVINO toolkit which includes OpenCV is used for the inferencing engine to develop the AI model and Python for coding. The main script executing at Level 3 which captures the frames is described below in Figure 10.

```
def process_frame(frame: VideoFrame) -> bool;
    for roi in frame.regions();
        for tensor in roi.tensors();
            if tensor.name() == 'detection';
                continue
            layer_name = tensor.layer_name()
            data = tensor.data()
            if layer_name == 'Feature_P2P_value';
                tensor.set_label(str(int(data[0] * 100)))
                continue
            if layer_name == 'anomaly';
                tensor.set_label("No" if data[1] > 0.5 else "Yes")
                continue
    return True
```

Figure 10 - Frame capture script

The function running is `process_frame()` which returns value `True` if the frame is ok and execution should continue. If the return value is `False`, the frame is dropped. We analyze the video frame for each of the regions in the frame, for each of the tensors in the region. We are not processing the detection results, so if it's a detection tensor we don't touch it but if it's a classification result, we take the last layer name. If this layer is `Feature_P2P_value`, then according to the description, the data is the P2P feature and we just need to multiply it by 100 according to the scaling requirements. If the layer is `anomaly`, then we have the probability, if normal or abnormal.

The Regul PLC is coded in a common programming language known as Ladder logic. Conditioned vibration signals are read through vibration sensor input card, samples at a configurable rate of 500ms and stored in FIFO tables which are read by the Level 2 computer, plotted and converted into a streamed video interpreted by the OpenVINO-based inference engine shown in Figure 9 which runs on the Level 3 system.

8 Conclusion

The development of HPC platform and AI toolkits have fuelled the development of more efficient and innovate ways for diagnostic and prognostic analysis. In this paper, we presented a proposed an approach to an equipment prognostic & availability management system where

feature engineering is required to take a central role [6] [7]. To date, literature demonstrating a how-to roadmap for feature engineering is very scarce, especially for industrial applications. Although deep learning models reduce the need for feature engineering, it is still the case that more specialized IIoT and AI models need to be better guided to perform on the most impactful features that can provide highest Return of Investment (ROI) of an AI implementation. It is to this end that this paper was written for those deciding where to start on their Digital Transformation journey.

Author Contributions: Writing-original draft P.D., writing-review and editing P.D., M.K.

Conflicts of Interest: Authors declare no conflict of interest.

References:

- [1] Niu, G. Data-Driven Technology for Engineering Systems Health Management: Design Approach, Feature Construction, Fault Diagnosis, Prognosis, Fusion and Decision; Springer: Beijing, China, 2016; ISBN 9789811020322.
- [2] Lei, Y. Intelligent Fault Diagnosis and Remaining Useful Life Prediction of Rotating Machinery; Elsevier BV: Amsterdam, The Netherlands, 2017.
- [3] Kim, S.; Lim, C.; Ham, S.-J.; Park, H.; Choi, J.-H. Tutorial for Prognostics and Health Management of Gears and Bearings: Advanced Signal Processing Technique.
- [4] Randall, R.B.; Antoni, J. Rolling element bearing diagnostics—A tutorial. *Mech. Syst. Signal Process.*
- [5] Peter Darveau, Wearable Air Quality Monitor, US Patent App. 14/162,897, 2015.
- [6] Goodman, D.; Hofmeister, J.P.; Szidarovszky, F. Prognostics and Health Management: A Practical Approach to Improve System Reliability Using Condition-Based Data; Wiley: New York City, NY, USA, 2019; ISBN 9781119356691.
- [7] Lee, J.; Wu, F.; Zhao, W.; Ghaffari, M.; Liao, L.; Siegel, D. Prognostics and health management design for rotary machinery systems—Reviews, methodology and applications. *Mech. Syst. Signal Process.* 2014