Proceedings of the 6th European Conference of the Prognostics and Health Management Society 2021 - ISBN – 978-1-936263-34-9

# Real-time Diagnosis Of Physical Failures Using Causation-based AI

Navid Zaman[1], Evan Apostolou[2], Yan Li[3] and Paddy Conroy[4]

[1] *PHM Technology, Melbourne, VIC, 3068, Australia*
*navid.zaman@phmtechnology.com*

## ABSTRACT

Every system in existence is prone to failure and analysis and early detection of said failures (for Predictive Maintenance) is becoming a crucial aspect of modern systems design. Most catastrophic issues start from the smallest parts of a component within the system (physical failures) and minute changes to certain sensor readings from this level may indicate that an incipient failure will occur. Much of this information and system knowledge is often captured during typical Reliability, Availability and Maintainability (RAM) activities but is not often re-purposed for diagnostics. Recent endeavors have been made to utilize AI-based calculations to analyze such measurements to notify that an anomaly is observed, however the nature of such correlation-based methods have a limit both on the information on such events and the reliability of the analysis due to spurious correlation. In this paper, we present a novel strategy to 'catch' failures before they happen using a combination of both correlation and causation, i.e. a causation-based AI and demonstrate its advantage over 'classic' correlation based methodology.

## 1. INTRODUCTION

Physical failures are the usually the initial cause of system disruptions in complex structures. These often include wear and tear, fatigue, cracks and so on which eventually cause the loss of the function of the component. These effects are often detected by symptom sensors such as vibration and temperature sensors placed upon certain parts. Such symptoms allow us to determine the possibility of certain failures before their occurrence (incipient failures). This means that signals from these sensors must be processed in real-time and the quantity of data to be analysed begs the use of machine learning techniques. Using clever correlation, these methods are able to detect anomalies and novelties thus are suitable to alert users of symptoms appearing within components. However, correlation will only estimate the likelihood of a symptom occurring but not the root cause or path to and from a fault. This paper outlines a causation-based AI approach, which takes into account the domain knowledge involving the dynamics of failure propagation whilst taking advantage of the computing power of machine learning and AI.



## 2. DEFINITION OF PHYSICAL FAILURES

### 2.1. Physical and functional failures

Broadly speaking all failures have some basis in the physical realm. The distinction that is made in this paper is regarding separating the physical phase of failure from the functional phase of failure for the purpose of diagnosis. Whereby physical failures may be detected based on observations arising from within the failing item pre-functional failure and functional failures may be detected based on functional changes to the system post-functional failure. Or in other words:

- Physical failures are referring to the failure dynamics internal to the local item experiencing the failure (e.g. the mechanism of failure is occurring on an item and that is resulting in a fault on that item).
- Functional failures are an evolution/continuation of a physical failure whereby the local item's ability to function is now disrupted.

Note that the principles of physical failure detection remain relevant post-functional failure, however the emphasis should be on pre-functional failure in order to place the potential diagnosis before failure effects begin to manifest.

### 2.2. P-F intervals as a method of identifying physical failures

This interpretation of failure progression shares much in common with the Potential to Functional (P-F) interval model elaborated on in Reliability-Centered Maintenance (RCM) 2 (Moubray, 1997). The P-F time interval plots the failure initiation point, the potential failure point (the theoretical point at which a failure can be detected), and the functional failure point (where the function of the item/system is degraded). The P-F interval or curve is for the most part theoretical as there is variability in both the P and F points. The potential failure point varies according to condition monitoring method (Bengtsson, 2007) and the functional failure point varies ac-

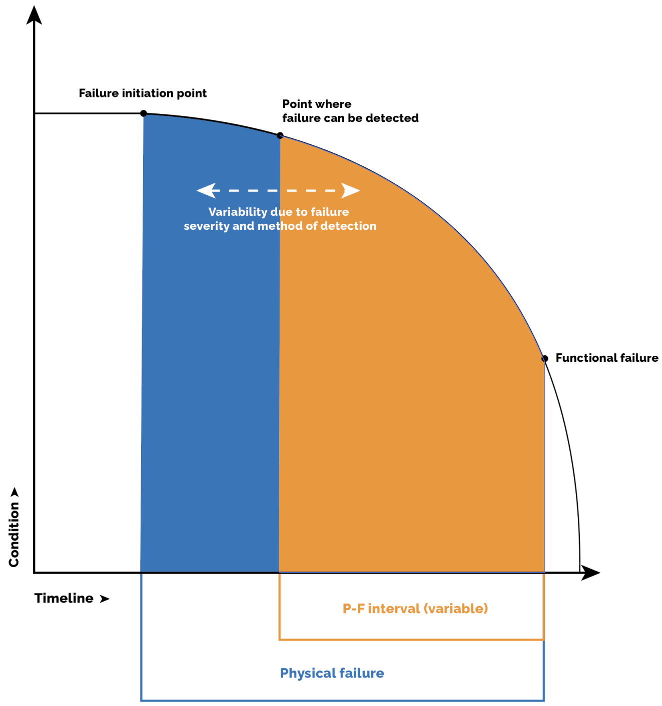



Figure 1. Typical P-F interval

cording to a variety of factors such as operating context and item variability (Dunn, n.d.). However, whilst the unique, specific characteristics of a P-F curve are not easily assessable, the general estimation and presence of the curve per failure is a key characteristic in determining viability of a prospective physical failure diagnosis approach as the presence of a large P-F curve allows:

1 The use of physical failure sensing on failures pre-functional failure
2 The mitigation of severe failure effects due to enough time between failure diagnosis and functional failure

As per RCM 2 (Moubray, 1997) the criteria for determining applicability of on-condition maintenance is:

1 A clear potential failure condition
2 The P-F interval is reasonably consistent
3 It is practical to monitor at intervals less that the P-F interval
4 The net P-F interval is long enough to be of use

With the application of real-time sensing, criteria 2 and 3 are practically negated, and the net P-F interval becomes synonymous with the P-F interval itself (assuming detection at first possible opportunity). Identification of clear failures is discussed further in terms of modelling (in section 3) and actual diagnosis (in section 4). In order to decently estimate a P-F interval, Moubray suggests two viable options. Firstly, testing and simulation of failures and secondly, educated judgement from those with knowledge of the system and its failure. If the P-F interval can be established to be of decently large size, then it should be considered a candidate for physical failure diagnosis.

## 3. Modeling of Physical Failures

### 3.1. RAM and diagnostic engineering roles

In order to cross the technical bridge between RAM and diagnostic engineering activities, a framework of understanding failures needs to be established that is consistent and serves the purposes of both practitioners.

The specific interests of RAMS and diagnostic engineers overlap in terms of their requirement to understand failures. The RAMS engineer needs to understand failures in terms of their broad statistical occurrence and consequences on the system (e.g. for use in Failure Modes, Effects and Criticality Analysis (FMECA) and reliability analysis). The diagnostics engineer needs to understand failures in order to detect and/or predict specific instances of failure. Whilst the use cases are different, ensuring a common paradigm to understand failures allows information developed in the RAMS space to be utilized in diagnostics as well as insights derived during diagnostics to be fed back to RAMS personnel, ultimately benefiting both roles.

### 3.2. Failure causes

A cause is the abnormal state of input, loading or environment that leads to the degradation of an item (Rudov-Clark & Stecki, n.d.). A cause can relate to design, manufacture, environmental, operational or maintenance actions or an input flow that exceeds specified limits. Causes are a key consideration when mitigating risk, however in every instance cannot be avoided entirely, as many are tied to the inherent operation of the system. As a result diagnostics and prognostics of the failure progressing post-cause is a viable additional mitigation method.

### 3.3. Failure mechanisms

Failure mechanisms are the physical process through which causes act to result in failure. As a mechanism is a process resulting in degradation (the fault) they may be the first observable occurrence in the failure path that can confirm a failure is happening.

### 3.4. Failure faults

A failure fault is the damaged or degraded state of the item that renders it unable to perform its function. This strict definition of a fault would indicate that once a fault is present then the functional failure associated with it will be observed through loss of function. Realistically the mechanism-fault-functional failure process is a progressive one in which the item may begin degrading due the mechanisms action while the function of the item is not disrupted (to an observable degree). Faults as they begin to manifest progressively impact item (and system) function. This paper is focused on the detection of failures prior to their functional manifestation. Detection of failures prior to functional failure requires some observable manifestation of failures. Faults themselves may be difficult to observe in their early stages. As such symptoms of those faults are used as secondary indicators of the failure.

### 3.5. Physical failure modelling in context of condition monitoring

The above cause-mechanism-fault framework of understanding failure development allows RAM engineers an optimal method of conducting analysis that can ultimately be used in specific failure mitigation activities. The framework also achieves the condition monitoring criteria of clear failure identification. As a result the implementation of a consistent taxonomy and structure of failure analysis benefits both parties as the knowledge of RAM engineers is fed directly into the diagnostics space. The proposed physical failure modelling framework is reconciled with the P-F model of incipient failures as follows (as per figure 2). The physical failure path is the entirety of failure from cause through to fault but prior to the functional failure. Functional failure is here defined as the local functional failure as well as any up or down stream effects on other items that the failure may have.

## 4. Symptoms as Anomalies

### 4.1. Physical Symptoms and Sensors

Physical events that occur as a result of a failure (fault) are called physical symptoms, which can be sensed by the physical sensors attached to them, aiding in detection and diagnosis.

Symptoms such as vibration, heat, and noise, can be interpreted as being binary, meaning that either the symptom occurs or not, which indicates the possibility of utilizing anomaly detection algorithms in the machine learning context to analyse these attached sensor signals (time series data) so as to detect possible failure faults. Whenever a symptom occurs in the system, a well-performed anomaly detection algorithm should be able to detect the existence of an anomaly by analysing the real-time data of the sensor measuring this specific symptom. Further, an alert could be triggered to inform the responsible parties to take action and avoid more severe failures.

Anomalies can be defined under various types or taxonomies such as contextual (dependent on operating condition or previous states), point (novelties or 'odd one out') and group (more akin to clustering, several points belonging to an 'anomalous group'). The taxonomy tackled in this paper is the contextual kind, since the method proposed takes into consideration the current operation mode and/or loading of the system when determining anomalies and single samples will consist of a window or several points, thus temporal effects are part of the features that are input.

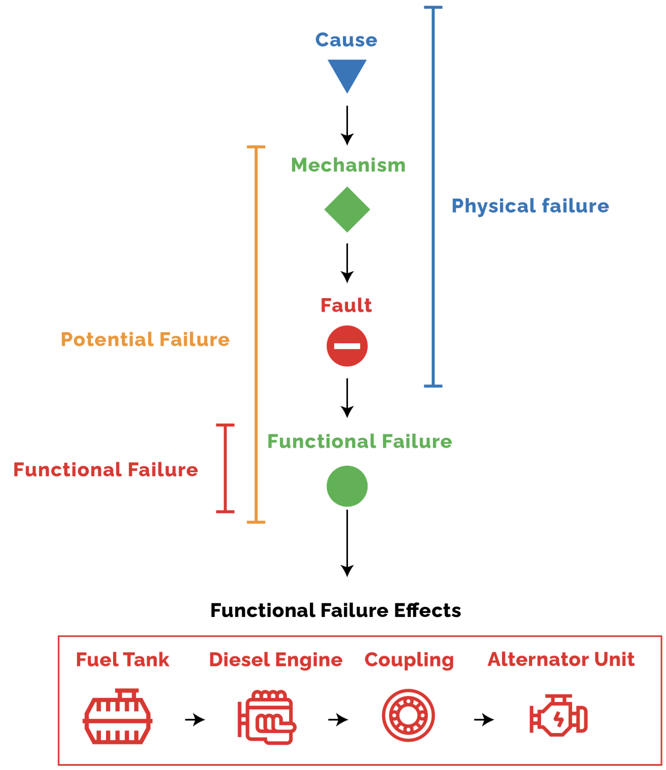



Figure 2. Failure path and the where it corresponds to P-F interval

### 4.2. Anomaly Detection In General

In general, anomaly detection or outlier detection aims to detect rare observations that are different from the majority of the sensor dataset (for example, in Figure 3, the black points (anomalies) are far away from the green points (normal points)), which is divided into three categories: supervised, semi-supervised, and unsupervised. If the training data consists of labels that indicate normal or abnormal, a supervised classification model can be trained on this labeled and imbalanced dataset and make predictions on the unknown data.

While training on dataset consisting only of normal observations, semi-supervised techniques define anomalies as observations deviating from the distribution of the training data. Unsupervised algorithms train on unlabeled data containing both normal and abnormal observations, which is taken under the assumption that anomalies are located in low-density regions in the data. Therefore, any new observations that are not located in high-density regions are considered anomalies.

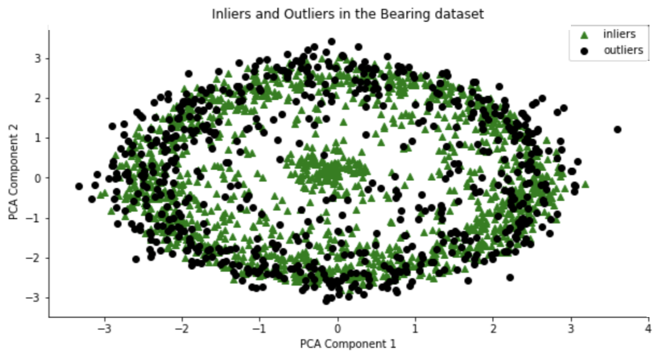



Figure 3. Anomaly Detection

The general idea is that an anomaly score is computed based on one or multiple measurements and the anomalies are determined based on a certain threshold, and if the anomaly score of observation is higher than this threshold, it is considered as an anomaly, otherwise, it is considered as a normal point. This threshold can be either pre-defined or computed from the data.

### 4.3. Individual Anomaly Detection Methods

Individual anomaly detection methods are commonly used to solve anomaly detection tasks, with which a single model is trained and the labels (normal or abnormal) of new observations are predicted using this trained model.

In the case of anomaly labels stated in the data itself, we can use methods such as Support Vector Machines (SVM) (Platt, 1999), a min-max based hyperplane and Gradient Boosting (Friedman, 2001), a tree-booster. For SVM, C, the regularization parameter which determines the amount of 'looseness' of the hyperplane and gamma are important. The kernel is common to keep to RBF (Radial Basis Function). For Gradient Boosting, as a tree method, parameters such as max depth and number of estimators are worth tuning.

Some of the these methods are proximity-based, for example, K-Nearest Neighbors (KNN) (Angiulli & Pizzuti, 2002) detect anomalies based on distance; the parameters that would be tuned are commonly the number of nearest neighbors to consider with regards to the distance. Local Outlier Factor (Breunig, Kriegel, Ng, & Sander, 2000) is based on local density and the number of neighbours within a certain density are tuned as parameters. Local Correlation Integral (LOCI) (Papadimitriou, Kitagawa, Gibbons, & Faloutsos, 2003) are based on clusters, also including micro-clusters, the diameter of clusters and distances between clusters.

Angle-base Outlier Detection (ABOD) (Kriegel & Zimek, 2008) is a probabilistic model which views the variance of an observation's weighted cosine scores to all neighbors as the anomaly score.

Principal Component Analysis (PCA) (Aggarwal, 2017), a

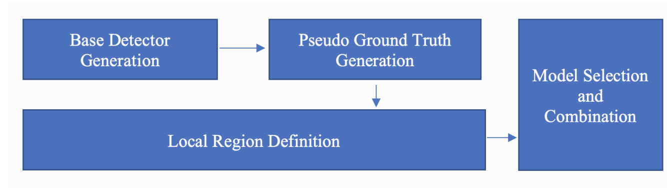



Figure 4. LSCP Workflow

linear dimensionality reduction technique, can also be used in detecting anomalies (called PCA detector in this paper), where the sum of the projected distance of a sample on all eigenvectors is utilised as the anomaly score. Anomalies tend to have higher scores.

### 4.4. Ensemble and Combination Methods

In contrast to individual methods, outlier ensemble methods can obtain more robust anomalies by combining the results from the different algorithm executions (Aggarwal, 2017), with Isolation Forest, Feature Bagging, and Locally Selective Combination in Parallel Outlier Ensembles (LSCP) (Zhao, Nasrullah, Hryniewicki, & Li, 2019) as representatives. Detector combination is a sub-field of ensembles, which combines various anomaly detector outputs by averaging, maximizing, average of maximum, maximum of average.

Isolation Forest trains a forest of random trees on random subdatasets and a new observation is considered abnormal when these trees collectively produce shorter path lengths for it. As a tree method, parameters such as the number of estimators (number of trees) and depth (of each tree) are important to tune.

The idea of Feature Bagging is that it randomly selects a subset of features, separately trains different anomaly detectors on these sub-spaces, and then aggregates the results by abovementioned combination methods. The number of estimators used to 'bag' each prediction can make a significant difference for this method.

## 5. Workflow for Physical Failure Detection

### 5.1. Overall Workflow

Figure 5 shows the full process from start to finish for detecting and isolating physical failures. Firstly the domain knowledge is established via a physical failure model which encompasses the possible routes of failure. Subsequently a machine learning tool is selected accordingly and trained (if required, in some unsupervised cases this is not required) using the historical data containing previous incidents. The trained algorithm and physical model are both used to determine the path of failure.

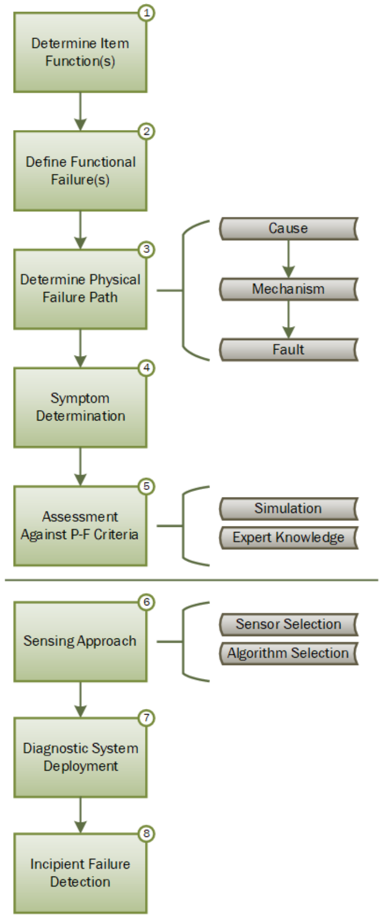



Figure 5. Workflow incorporating RAM and diagnostic activities

### 5.2. Case Study

#### 5.2.1. NASA Bearing Dataset

To demonstrate the capability of the strategy on physical failures, a NASA Bearings Dataset (Qiu, Lee, Lin, & Yu, 2006) from which a single bearing will be considered, sensed by a pair of accelerometers, each measuring x and y axes. Thus, the data consists of two columns which will be used for the diagnosis of physical failures in this system. It is stated that the roller element part of the bearing fails at the end of the test cycle, which takes over a month. The files used are vibration records separated by 10 minute intervals. In this section, a step-by-step description of the workflow in figure 5 is provided in the context of the NASA bearing dataset.

1 Determination of item failures is a typical systems engineering task whereby the item in question is defined by the functional outputs they provide / facilitate. For the bearing being analysed in this case study, it can be said that it has a functional flow of "allow rotational/angular velocity". By defining the functional flow of an item, it allows for the definition of the functional failure of the item and how the functional failure may propagate to impact other items / the system mode broadly. For example, the bearing in a system which functionally fails (no longer allows for angular velocity) will cause other items downstream to be unable to perform their function.

2 The definition of the functional flow of the bearing (allow angular velocity) will allow for functional failures being defined. Functional failures are used to describe how the item has failed from a functional perspective, and for this bearing it's functional failure is "allow angular velocity low". The "low" describes that the angular velocity output of the bearing is lower than the nominal or expected value. Similarly, "high" would be used to describe the angular velocity output of the bearing is higher than the nominal or expected value, but since this is not possible due to physical failures of the bearing, it is not defined as a functional failure in this case study.

3 The cause-mechanism-fault framework defined above is used to create the physical failure diagram for the bearing in this case study. Failure diagrams are developed at the most granular level of information applicable. In a bearing that is an inner ring, outer ring and rollers For the interest of this case study, only the roller element failure diagram will be shown in figure 6.
Cause, mechanisms and faults are represented by their icons as described earlier, and each of them are populated in the failure diagram as they are common failure concepts of roller elements in bearings - the physics of failure will depend on loading and operating conditions, so common physical failures are used in this failure diagram (SKF, n.d.).

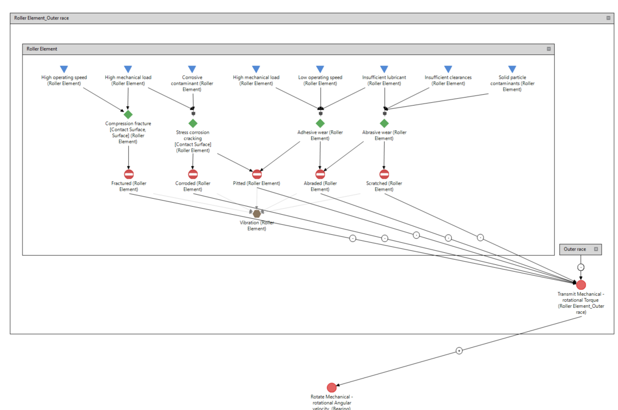


Figure 6. Bearing Failure Diagram

4 The brown hexagon icon in the failure diagram is a symptom. The symptom is used to describe what physical observations can be manifested as a result of a fault occurring - these are used to determine when failures have occurred in an item. For this case study, since accelerometer sensors are placed on the bearing, it is appropriate

to place the vibration symptom in the failure diagram to describe that this will be measured and observed, and that these faults cause vibration in the bearing to increase when/if they occur.

5 To determine the suitability of a symptom for sensing, assessing the length of the P-F interval is required. Based on the loading characteristics (bearings are underloaded) and short timeframe of data, it can be assumed that there is a sufficient P-F interval for the failures defined in the failure diagram in figure 6 (Qiu et al., 2006).

6 Since the vibration symptom is attached to all faults in the failure diagram, it can be said that 100% of the failures of interest are covered by the accelerometer sensors.

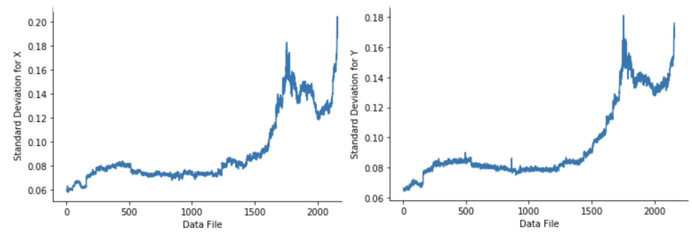
Data is split into training, validation (used to fine tune algorithms) and testing where the latter is completely unseen by the algorithms until prediction is attempted. Since it is already stated that a run-to-failure test is done on the bearing, the final few (656 files) are labelled as failure/anomaly and readings before that are essentially healthy (1500 files are used as healthy). Using this information, the data is labelled as anomalous or not and fed into the training/testing pipeline to prepare the algorithms. The data was first feature engineered and then presented to several supervised and unsupervised algorithms. The metrics used for supervised include f1-score, accuracy and ROC-AUC (Area Under Receiver Operating Curve); in addition to these, for unsupervised we used EM (Excess-Mass, the more the better) and MV (Mass-Volume, the less the better) (Goix, 2016). A summary of the results and label details are provided in table 1 and 2 respectively. The results table compares both supervised and unsupervised methods for this problem.

7 The algorithm will be trained and ready for operation. Sensors will be collected via an IoT platform, which is able to organize and translate values from digital readings to the correct format. This will be transferred to the algorithm operating either within the platform itself or running in the cloud, as a server application. In the case that it is the latter, the platform will communicate through high-speed protocols such as MQTT or otherwise to send sensor readings ASAP for analysis.

In this case study, we emulated the real-time sensors using data from the testing set to simulate a operational scenario.

8 Vibrations on the bearing are visualized in 7 via the standard deviation obtained from the bearing dataset. These images show a clear change when the bearing is failing.

As real-time information (emulated here) is inferred upon, we use the predicted results and the failure diagram together to determine that the roller element had failed as well as trace the possible faults back to their root cause.




(a) Vibration deviation for x-axis (b) Vibration deviation for y-axis

Figure 7. The vibration signals' standard deviation for the bearing

Table 1. Summary of results. Full table in Appendix, table 3

| Best Algorithm | Feature Extractor | ROC-AUC Score | EM |
|---|---|---|---|
| Gradient Boost (supervised) | Statistical | 99.79% | N/A |
| PCA Detector (unsupervised) | Statistical | 97.29% | 1.77 |

Table 2. Table for the labels in the bearing data split up

| y split | Total in split | Labels | Distribution |
|---|---|---|---|
| y_train | 943 | [0, 1] | [656, 287] |
| y_val | 566 | [0, 1] | [394, 172] |
| y_test | 647 | [0, 1] | [450, 197] |

## 6. Limitations of this method

This method is highly effective when domain knowledge about the system/component is known and can be modelled into a failure diagram equivalent. Training data is also a requirement in order to apply the ML methods mentioned. Real-time failure analysis is the goal for this paper - to predict anomalies and determine incipient failures, not analysis on static data. The case study was used to demonstrate the capability, however in operation, the vibration data are expected to be received from a streaming engine with the algorithms and domain information working as a server application.

## 7. Conclusion

To summarize, this paper has presented a method to perform physical failure detection and analysis, utilizing both causal information involving the physics of failure and correlation to achieve real-time estimation of faulty readings. The real time aspect is crucial in these situations since the goal is to 'catch' the failure before it affects the function of the component. Thus we not only predict an (expected, in this case) anomaly in the bearing but can also tell what part and possible causes involved.

## References

Aggarwal, C. C. (2017). *Outlier analysis*. Springer, Cham.

Angiulli, F., & Pizzuti, C. (2002). Fast outlier detection in high dimensional spaces. Springer.

Bengtsson, M. (2007). *On condition based maintenance and its implementation in industrial settings* (Unpublished doctoral dissertation). Mälardalen University Sweden.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). Lof: identifying density-based local outliers. ACM.

Dunn, S. (n.d.). *The pf interval – is it relevant in the world of big data?*

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5), 1189 – 1232. Retrieved from `https://doi.org/10.1214/aos/1013203451` doi: 10.1214/aos/1013203451

Goix, N. (2016). *How to evaluate the quality of unsupervised anomaly detection algorithms?*

Kriegel, H.-P., & Zimek, e. a., Aurthur. (2008). Angle-based outlier detection in high-dimensional data. ACM.

Moubray, J. (1997). *Reliability-centered maintenance 2nd edition*. Butterworth-Heinemann.

Papadimitriou, S., Kitagawa, H., Gibbons, P. B., & Faloutsos, C. (2003). Loci: fast outlier detection using the local correlation integral. IEEE.

Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in large margin classifiers* (pp. 61–74). MIT Press.

Qiu, H., Lee, J., Lin, J., & Yu, G. (2006). Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics. *Journal of Sound and Vibration*, *289*, 1066–1090.

Rudov-Clark, S. D., & Stecki, J. (n.d.). The language of fmea: on the effective use and reuse of fmea data. In *Aiac-13 thirteenth australian international aerospace congress.*

SKF. (n.d.). *Bearing damage and failure analysis.*

Zhao, Y., Nasrullah, Z., Hryniewicki, M. K., & Li, Z. (2019, May). LSCP: locally selective combination in parallel outlier ensembles. In *Proceedings of the 2019 SIAM international conference on data mining, SDM 2019* (pp. 585–593). Calgary, Canada. Retrieved from `https://doi.org/10.1137/1.9781611975673.66` doi: 10.1137/1.9781611975673.66

## 8. Appendix

Table 3. Table for comprehensive tests done

| NoScaler + Stat (Reducer) | | | Accuracy | | f1_score | | AUC | | EM | | MV | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Feature Extractor | Algorithm | X_val | X_test | X_val | X_test | X_val | X_test | X_val | X_test | X_val | X_test |
| Supervised | PCA | SVM | 0.62 | 0.62 | 0.63 | 0.63 | 0.62 | 0.63 | | | | |
| | PCA | GradientBoost | 0.69 | 0.70 | 0.59 | 0.60 | 0.57 | 0.57 | | | | |
| | Stat | SVM | 0.95 | 0.96 | 0.95 | 0.96 | 0.99 | 0.99 | | | | |
| | Stat | GradientBoost | 0.98 | 0.98 | 0.98 | 0.98 | 1.00 | 1.00 | | | | |
| Unsupervised | Stat | OCSVM (always predict 0) | 0.70 | 0.70 | 0.57 | 0.57 | 0.50 | 0.50 | 1.13 | 0.00 | 1.74E+11 | 8.63E+11 |
| | Stat | KNN | 0.70 | 0.70 | 0.57 | 0.57 | 0.66 | 0.70 | 0.00 | 0.01 | 2.56E+14 | 2.33E+12 |
| | Stat | Feature Bagging | 0.70 | 0.69 | 0.58 | 0.57 | 0.44 | 0.49 | 0.01 | 0.00 | 1.77E+12 | 9.77E+14 |
| | Stat | LOF | 0.69 | 0.70 | 0.57 | 0.58 | 0.47 | 0.53 | 0.00 | 0.02 | 1.02E+12 | 3.48E+15 |
| | Stat | Isolation Forest | 0.82 | 0.85 | 0.81 | 0.84 | 0.91 | 0.93 | 0.11 | 0.04 | 1.37E+12 | 6.38E+15 |
| | Stat | PCA_detector | 0.93 | 0.91 | 0.93 | 0.91 | 0.97 | 0.97 | 0.00 | 0.17 | 1.21E+15 | 7.22E+14 |
| | PCA | OCSVM | 0.67 | 0.68 | 0.59 | 0.61 | 0.56 | 0.52 | 1.77 | 7.03 | 0.05 | 0.03 |
| | PCA | KNN | 0.68 | 0.68 | 0.59 | 0.58 | 0.66 | 0.59 | 0.40 | 0.77 | 0.07 | 0.05 |
| | PCA | Feature Bagging | 0.69 | 0.69 | 0.57 | 0.57 | 0.57 | 0.58 | 0.08 | 0.01 | 0.28 | 2.62 |
| | PCA | LOF | 0.70 | 0.69 | 0.58 | 0.58 | 0.59 | 0.56 | 1.33 | 0.72 | 0.03 | 0.05 |
| | PCA | Isolation Forest | 0.66 | 0.67 | 0.55 | 0.57 | 0.51 | 0.55 | 0.01 | 0.07 | 2.03 | 0.35 |
| | PCA | PCA_detector (always predict 0) | 0.70 | 0.70 | 0.57 | 0.57 | 0.55 | 0.56 | 0.99 | 0.71 | 0.03 | 0.04 |

## Biographies

**N. Zaman** Navid Zaman is a master of electrical engineering graduate from the University of Melbourne, Australia since 2020, where he focused on signals, systems and control theory. He has interned at Outotec Ausmelt for a few months before joining PHM Technology as lead data scientist. He has co-authored a RAMS paper previously, centered around the causation-based AI tool, Syndrome Diagnostics.

**E. Apostolou** Evan Apostolou is currently working at PHM Technology as an Engineer. His primary focus in the company is engineering support for Syndrome Diagnostics and MADe. Since graduating from the University of Melbourne with a Master of Engineering (Mechanical), he has worked at PHM Technology, gaining experience and knowledge in the safety, RAMS and PHM industries. He previously worked at RUAG Australia, Bosch Australia and Airbus Helicopters Germany where his roles varied from materials engineering, maintenance and system design.

**Y. Li** Yan Li is a data scientist at PHM Technology, mainly focusing on the research and development of the Syndrome Diagnostics product. She has got her Master of Information Technology degree at the University of Melbourne, Australia since 2020. Before her master's study, she had been working for 6 years in the logistics and training industries and has a good understanding of customer needs for business software tools.

**P. Conroy** Paddy Conroy is a Senior Engineer and Research Lead at PHM Technology. He received a Bachelor degree of Aerospace Engineering through RMIT University, Australia. Paddy has 6+ years of experience in a range of engineering disciplines including aerospace, naval, land, and automotive across mechanical and electrical domains. His specific interests are the research and development of RAMS and diagnostic model-based engineering methodologies and the processes required to integrate them into design and sustainment activities.