Supplementary on: A physical-statistical framework on complex mechanical system fault isolation

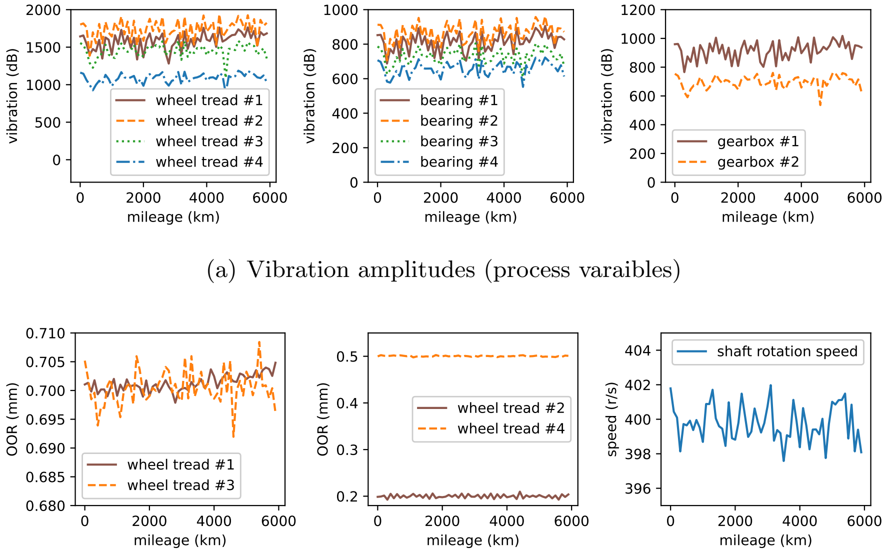
1 Details on the dataset

The *composite sensors #1–4* measure the original vibrations signals with high frequency. These original vibration signals are then sent to the *resonant demodulators #1–4* through the *cables #1–4* as follows:

- The *resonant demodulator #1* demodulates the vibration signals from *composite sensor #1* into the vibration amplitudes generated by gearbox #1, wheel #1, and bearing #1.
- The *resonant demodulator #2* demodulates the vibration signals from *composite sensor #2* into the vibration amplitudes generated by wheel #2 and bearing #2.
- The *resonant demodulator #3* demodulates the vibration signals from *composite sensor #3* into the vibration amplitudes generated by gearbox #2, wheel #3, and bearing #3.
- The *resonant demodulator #4* demodulates the vibration signals from *composite sensor #4* into the vibration amplitudes generated by wheel #4 and bearing #4.

In addition to vibration amplitudes, the resonant demodulator can obtain the OORs of wheel treads as well. Moreover, the *speed sensor* reads and stores the rotation speeds of shaft #1 and shaft #2. The original readings of vibration signals from *composite sensors #1–4* are not stored to reduce storage burden. Instead, the resonant demodulators resample the vibration amplitudes, the rotating speed, and the OOR measurements every 100 km and send the resampled measurements to the SCADA data center.

Taking the BG00 bogie as an example, we present a segment of SCADA data in Figure 1 from the power bogie BG00, which includes vibration amplitudes, shaft rotation speed, and OOR. We also present the mean and standard deviation (STD) for each variable in the dataset from the BG00 bogie in Table 1.



(a) Vibration amplitudes (process varaibles)

(b) OORs and rotation speed (covariates)

Figure 1: A segment of SCADA data from the power bogie BG00, which includes the vibration amplitudes, the shaft rotation speed, and the the out-of-roundnesses (OORs).

Table 1: Mean and standard deviation (STD) for each variable in the dataset from BG00 bogie.

Variable		Mean	STD
Vibration amplitude	wheel #1	1585	114.3
	wheel #2	1740	131.5
	wheel #3	1450	89.93
	wheel #4	1086	64.06
	bearing #1	812.6	53.84
	bearing #2	866.7	58.06
	bearing #3	730.5	44.4
	bearing #4	657.3	39.21
	gearbox #1	915.4	61.58
	gearbox #2	696.4	43.58
Out-of-roundness	wheel #1	0.7013	0.0014
	wheel #2	0.1999	0.0040
	wheel #3	0.7004	0.0030
	wheel #4	0.5003	0.0011
Speed		399.8	1.065
Mileage		2950	1746

2 Additional experiments

We conduct two additional experiments for validation.

First experiment:

We first conduct a statistical test to compare the performance of the proposed methods and two benchmarks. Using different regression model, including the linear, quadratic, and cubic spline, we have three types of our proposed model. For the benchmark, the first one (denoted as PCACP) employs PCA to compute anomaly scores, applies the Q-statistic for anomaly detection, and uses a contribution plot to identify faulty components. The second benchmark (denoted as DAECP) follows the same anomaly detection pipeline as PCACP but replaces PCA with a DAE for anomaly score computation.

To test the statistical significance of these $K = 5$ methods (three types of our method and two benchmarks) based on their performance in terms of F1-scores on $N = 4$ datasets from four bogies, we use the Friedman test and Conover post-hoc test following [1]. We first apply the Friedman test to determine whether there are significant differences in performance among all methods. The null hypothesis H_0 states that all methods have equal performance, while the alternative hypothesis H_1 is that at least one method performs significantly differently from the others:

$$\begin{aligned} H_0: &\quad R_1 = R_2 = \cdots = R_5, \\ H_1: &\quad \exists k \neq j \quad \text{such that} \quad R_k \neq R_j, \end{aligned}$$

where R_k represents the average rank of method k across N datasets, with $k \in [K]$ and $j \in [K]$. The average rank of method k across all N datasets is estimated as:

$$\widehat{R}_k = \frac{1}{N} \sum_{i=1}^{N} r_{i,k},$$

where $r_{i,k}$ denotes the rank of method k on dataset i for $k \in [K]$ and $i \in [N]$. The best-performing method receives rank 1, the second-best receives rank 2, and so on. In case of ties, the average rank is assigned. Given the estimated ranks, the Friedman test statistic is computed as:

$$\mathcal{F} = \frac{12N}{K(K+1)} \sum_{k=1}^{K} \widehat{R}_k^2 - 3N(K+1).$$

Under the null hypothesis, the test statistic $\mathcal{F}$ asymptotically follows a chi-square distribution with $(K-1)$ degrees of freedom, i.e., $\mathcal{F} \sim \chi^2(K-1)$. If the p-value of the test statistic is smaller than a given significance level α, the null hypothesis is rejected; otherwise, it is not rejected.

When the Friedman test rejects the null hypothesis, it indicates that at least one model performs differently from the others, but it does not reveal which specific pairs of models differ. To identify these pairwise differences, we employ the Conover post-hoc analysis following [1]. The test statistic for comparing models k and j is given by:

$$\mathcal{T}_{k,j} = \frac{\left|\widehat{R}_k - \widehat{R}_j\right|}{\sqrt{\frac{K(K+1)}{6N}}}.$$

Under the null hypothesis that there are no differences between any pairs of models, the statistic $\mathcal{T}_{k,j}$ approximately follows a normal distribution. For each pair (k, j), if the p-value corresponding to $\mathcal{T}_{k,j}$ is smaller than a given significance level, we conclude that there is a statistically significant difference in performance between models k and j.

We apply the Friedman test to the performance metrics from the proposed and benchmark methods, based on their results in Table IV of the revised manuscript. The Friedman test yields a small p-value (0.0155), indicating sufficient significance to reject the null hypothesis and confirming that there are significant differences among the three proposed methods and the two benchmarks. We further conduct a Conover post-hoc analysis to determine which specific methods differ. The pairwise p-values among all the methods are presented in Table 2. Table 2 indicates significant differences between the benchmark DAECP and our methods. Due to page limits, these results will also be publicly available in the supplementary file on GitHub (https://github.com/Yan9564/SCADA.git).

Table 2: The p-values of pairwise comparisons between the competing approaches.

Method	Proposed (C)	Proposed (Q)	Proposed (L)	PCACP	DAECP
Proposed (C)	1	0.0735	0.7370	0.0356	0.0153
Proposed (Q)	-	1	0.1345	0.7058	0.0003
Proposed (L)	-	-	1	0.4785	0.0076
PCACP	-	-	-	1	0.0327
DAECP	-	-	-	-	1

"Proposed (C)", "Proposed (Q)", and "Proposed (L)" are the proposed methods with cubic, quadratic, and linear splines, respectively. The first benchmark (denoted as **PCACP**) uses PCA to obtain the anomaly scores, the Q-statistic to detect anomalies, and a contribution plot to isolate faulty components. The second benchmark (denoted as **DAECP**) is similar to **PCACP**, except that it uses a DAE to obtain the anomaly scores.

Second experiment:

We then compare the performance of our anomaly detection method with the two benchmarks using the average running length (ARL_0) of the false alarm. The first benchmark (denoted as PCACP) employs PCA to compute anomaly scores, applies the Q-statistic for anomaly detection, and uses a contribution plot to identify faulty components. The second benchmark (denoted as DAECP) follows the same anomaly detection pipeline as PCACP but replaces PCA with a DAE for anomaly score computation. The first benchmark is used as in the previous version of our manuscript, and the second benchmark is newly added in response to comment #9 from you. The ARL_0 values of our method, PCACP, and DAECP are 34863.08, 8031.91, and 103.32, respectively, indicating that the proposed anomaly detection method performs satisfactorily.

References

[1] Ahmed, I., Galoppo, T., Hu, X., & Ding, Y. (2021). Graph regularized autoencoder and its application in unsupervised anomaly detection. IEEE transactions on pattern analysis and machine intelligence, 44(8), 4110-4124.