Postprocessing of Autoencoder Reconstruction Error for Detection and Diagnostics of Faults in Infrequently-driven Ground Vehicles

Matthew Moon[1], Ethan Kohrt[2], Michael Thurston[3], and Nenad Nenadic[4]

[1,2,3,4] *Rochester Institute of Technology, Rochester, NY, 14623, USA*
memgis@rit.edu
eakgis@rit.edu
mgtasp@rit.edu
nxnasp@rit.edu

ABSTRACT

We investigated the detection and classification of engine and transmission faults in infrequently-driven ground vehicles using data-driven methods based on neural network autoencoders. The data came from seventeen vehicles, each with an engine-related or a transmission-related maintenance event. The vehicles had months to years of sensor controller area network (CAN) bus data sampled at 1Hz. Separate autoencoder models were trained for each vehicle to improve detection sensitivity. The paper investigates several condition indicators (CIs) derived from autoencoder reconstruction error, each computed from a sequence of the reconstruction's mean absolute error (MAE). These CIs were compared using a performance metric computed as the area under the Pareto front with respect to normalized detection horizon and normalized baseline-relative CI margin. A novel detection procedure, *consistent detection*, effectively filtered out short-duration isolated spikes, likely false positives, while also increasing sensitivity to more plausible anomalies. In addition, the initial development of data-driven diagnostics, based on a novel approach of classifying full reconstruction error vectors associated with the fault state, showed promise but failed our robustness checks.

1. INTRODUCTION

Neural network autoencoders are common models for anomaly detection. An autoencoder is trained to minimize the error between the input data and the model output, called reconstruction error. When the autoencoder is trained only on data representing the nominal state of some system, the model is expected to output low reconstruction error for new data from the nominal state, and high reconstruction error for data from anomalous states. The reconstruction error can be used to calculate an "anomaly score," which is compared against some threshold to decide whether a data sample is an anomaly.

Mean Squared Error (MSE) and Mean Absolute Error (MAE) are frequently used as anomaly scores (Adkisson, Kimmell, Gupta, & Abdelsalam, 2021; Peixoto et al., 2023; Lachekhab, Benzaoui, Tadjer, Bensmaine, & Hamma, 2024; S. Ryu et al., 2023; Tziolas et al., 2022; Tian, Liboni, & Capretz, 2022; Lagazo, de Vera, Coronel, Jimenez, & Gatmaitan, 2021). In (Tziolas et al., 2022) the authors commented that MSE was prone to outliers in their work, and therefore preferred MAE instead. In MAE and MSE, the contribution of each channel in the anomaly score is weighted equally. This is sometimes not desired; the authors in (S. Ryu et al., 2023) observed that their reconstruction error was concentrated in a few channels, but the influence of these channels was diluted due to the high dimensionality of their data. To address this, they used only the top-k channels with the highest error when calculating MSE. Root mean squared error (RMSE) is also sometimes used as anomaly score (Park, Marco, Shin, & Bang, 2019; Reddy, Sarkar, Venugopalan, & Giering, 2016).

Mahalanobis distance is another common anomaly score (Thill, Konen, Wang, & Bäck, 2021; Zhang, Hu, & Yang, 2022; Ahmad, Styp-Rekowski, Nedelkoski, & Kao, 2021). In this method, a multivariate Gaussian distribution is estimated based on the raw reconstruction error of the healthy training data. Then, for a given data sample, the anomaly score is calculated as the Mahalanobis distance between that sample's reconstruction error and the distribution of reconstruction errors seen in the training set. Lagazo et al. (Lagazo et al., 2021) used a similar approach but instead calculated the log-likelihood that an error sample came from the training error distribution. Liang et al. (Liang, Knutsen, Vanem, Zhang, & Æsøy, 2023) utilized Sequential Probability Ratio Test (SPRT) to analyze reconstruction error samples and determine whether they more closely aligned with an approx-

imated nominal or anomalous error distribution. Yan et al. (Yan, Guo, gong, & Li, 2016) showed that even the magnitude of the hidden layer outputs could be successfully applied as an anomaly score.

Filters are sometimes applied to smooth the anomaly scores when the data is a time-series. The simplest case of a sliding mean is used to mitigate false positives from point anomalies in (Zhang et al., 2022; Tian et al., 2022).

A detection threshold is necessary to determine whether an anomaly score represents a nominal or anomalous data sample. When labeled data is available, a threshold is often chosen empirically to maximize desired metrics like F1-score, or to minimize either false positives or false negatives based on requirements of the domain (Lachekhab et al., 2024; Shvetsova, Bakker, Fedulova, Schulz, & Dylov, 2021; Mallak & Fathi, 2021; Tziolas et al., 2022). It is also common to calculate anomaly scores of the nominal data and set the threshold above those scores to mitigate false positives. Since autoencoders are trained and validated on nominal data, (Peixoto et al., 2023) sets the threshold to the maximum anomaly score in the training set. To prevent single large anomaly scores from skewing the threshold, (Adkisson et al., 2021) set the threshold as the average of the top 5 validation scores. Ryu et al. (S. Ryu et al., 2023) places the threshold at 200% of the mean validation score, and (Connelly, Zaidi, & McLernon, 2023) sets multiple thresholds at increasing standard deviations from the mean validation score to indicate increasing severity levels. Tuli et al. (Tuli, Casale, & Jennings, 2022) compared two other thresholding techniques in their approach; Peak Over Threshold (POT) (Siffer, Fouque, Termier, & Largouet, 2017) and Annual Maximum (AM) (Bezak, Brilly, & Šraj, 2014).

Some PHM diagnostic approaches do not use an intermediate anomaly detector (Deng, Wang, Tang, Huang, & Zhu, 2021; Kreuzer & Kellermann, 2023; Shen, Wang, Fu, & Xiong, 2023), instead opting for a classifier that outputs "normal" or a particular fault type. But an autoencoder can also be paired with a classifier model to perform fault mode diagnostics (Michau, Hu, Palmé, & Fink, 2019). Some approaches pass data samples through an autoencoder to first detect an anomaly, then pass the same anomalous data to a trained classifier to predict the fault type (Zhang et al., 2022; Mallak & Fathi, 2021; Park et al., 2019). Latent features of the autoencoder may also be used in the classifier. In (G. Ryu & Seong, 2023), though the model was a transformer trained to predict masked data rather than an autoencoder, latent features from the model were extracted and a K-Nearest-Neighbors approach was used to classify fault types. Shao et al. (Shao, Jiang, Wang, & Zhao, 2017) did not perform anomaly detection, but demonstrated that the features learned by denoising and contractive autoencoders resulted in better accuracy than existing methods for fault mode classification.

Autoencoder reconstruction error may also contain useful information for diagnostics. Reddy et al. (Reddy et al., 2016) implemented autoencoder anomaly detection on aircraft flight data, and demonstrated that different fault modes could be distinguished from one another based on the distribution of reconstruction error across signals. Krishnan et al. (Krishnan et al., 2024), Hsu et al. (Hsu, Frusque, & Fink, 2023) and Torabi et al.(Torabi, Mirtaheri, & Greco, 2023) set a separate threshold for each channel of error, corresponding to each input signal, to give information to an operator about which specific signals are responsible for an anomaly. Vuong et al. (Vuong, Giduthuri, Lim, Tan, & Ramasamy, 2024) visualized the error contribution of each signal and found that the contribution from some signals was far greater than other signals, and the predominant signals varied between fault instances, but there was no attempt to establish a relationship between signal contributions and fault mode. In the same vein, de Pater et al. (de Pater & Mitici, 2023) observed that only a select few signals contributed error that corresponded with the fault, and created a stronger detector by ignoring the error contributions of the other signals. Hsu et al. (Hsu et al., 2023) used Silhouette score to show that samples of reconstruction error do cluster somewhat according to their fault mode, though the clusters became less distinct as the faults developed into a more severe state. However, none of these methods directly involve classification of autoencoder reconstruction error.

Our prior work (Kohrt et al., 2024) focused on preprocessing and modeling. It identified information-carrying signals, and experimented with the length of the observation period. It explored different AE topologies for detecting engine faults in ground vehicles, trying to achieve the best anomaly detection performance. The condition indicator was the mean absolute error (MAE), averaged over several observations. That approach often suffices when the available signals carry the information on the failure in progress. When the signals are noisy, the problem gets worse. One approach is to investigate additional anomaly detection models, and propose a novel model. However, at present, high-capacity models are sufficiently expressive (TODO cite). We observed this in our work, too. The detection capability of different of autoencoders with different topologies (fully-connected, 1D CNN, transformers) was quite similar.

In this paper we took a different approach. Instead of trying to improve the models, we attempted to extract better anomaly detection performance by operating on the autoencoder error beyond MAE and MSE. Furthermore, we attempted to investigate a path to diagnostics, by operating on the reconstruction errors from the autoencoders. Specifically, we offer an analysis of various condition indicators (CIs), which for our purposes are anomaly scores on the reconstruction error of fixed-length windows of multivariate time-series sensor data. The CIs we investigate are all derived from local MAE like (Peixoto et al., 2023; Tziolas et al., 2022). This analysis in-

cludes considerations of the trade-off between detection horizon and detection confidence, the latter measured in terms of deviation from baseline behavior. We evaluate CI strength by measuring the area bounded by the Pareto front with respect to normalization of these two detection metrics.

Additionally, we introduce a new detection routine deemed "consistent detection". Consistent detection is a methodology intended to reduce the impact of isolated outliers and increase confidence that detections found are true positives. This new detector also allows us to reduce the sensitivity of the detection threshold to baseline outliers.

Finally, we demonstrate the feasibility of using autoencoder reconstruction error for fault-classification diagnostics, including applications to a real-time system. Our classification systems are based on only the data associated with anomalies, unlike approaches that further consider nominal data such as (Reddy et al., 2016).

To the best of our knowledge, this paper is the first to:

- Apply log-likelihood to the reconstruction error, which improved the anomaly detection over typically used MAE / MSE.
- Combined margin and detection horizon in the form of Pareto front, and then used the area to under the curve of the Pareto front for the metric. Furthermore, we applied a new nonlinear transformation to better normalize metrics among assets (Section 2.3.1).
- Formulate criteria for avoiding false alarms by demanding that the anomaly persists or reoccurs (see Figure 2).
- Use reconstruction error as features in a classifier for diagnostics.

2. ANOMALY DETECTION

To detect faults from vehicle sensor data, we created a model of baseline data, and looked for unprecedented model behavior in the time leading up to a known fault. The boundary between the baseline period and the fault window was determined by inspection of maintenance history by a subject matter expert (SME). We chose neural-network autoencoders for our model, which seek to compress and then reconstruct some window of sensor data. More specifically, we used a transformer-based architecture we call "TFAE-Sym" as described in (Kohrt et al., 2024). Data windows that are similar to the training data should yield low reconstruction error, and anomalous windows should yield high error.

In our prior work (Kohrt et al., 2024), we detected anomalies in the multivariate reconstruction error by finding when the running mean of absolute baseline error exceeded some threshold. Under this scheme, four vehicles exhibited detection. Table 1 lists the detection margin and detection horizon (measured in engine-on hours) for the vehicles under investigation. In this section, we describe a formalization of this procedure that allows for alternative error processing and detection procedures.

Table 1. Detection characteristics in the prior postprocessing procedure with a CI of Mean with $K = 20$ on idle-only data.

IDs	Fault Type	Detection Horizon	Margin
E01	Cooling Fault	33.6 h	24%
E02	Cooling Fault	-	0%
E03	Thermostat Failure	30.2 h	53%
E04	Fuel Injector Defect	21.0 h	225%
E05	Fuel Injector Defect	-	0%
E06	Coolant Leak	213.5 h	82%
E07	Coolant Leak	-	0%
E08	Coolant Leak	-	0%

2.1. Condition Indicators

Starting with a mathematical framing, an autoencoder is a function ψ on $\mathbb{R}^{N\times S}$ which should approximate the identity on baseline data and hopefully yield divergent behavior for anomalous data. For a given two-dimensional autoencoder input X_i, we consider its signed reconstruction error

$$\begin{aligned} E_i &= [e_{n,s}]_i \\ E_i &= \widehat{X}_i - X_i = \psi(X_i) - X_i \end{aligned} \tag{1}$$

where $\psi(\cdot)$ denotes the autoencoder function.

The reconstruction error E_i has the same dimension as the input X_i, which is a $N \times S$ matrix, where N represents the number of measurements (i.e. the observation period), S is the number of signals, and the elements of the reconstruction matrix are denoted by $e_{n,s}$ (see Figure 1). For our analysis, we kept N fixed at 30.

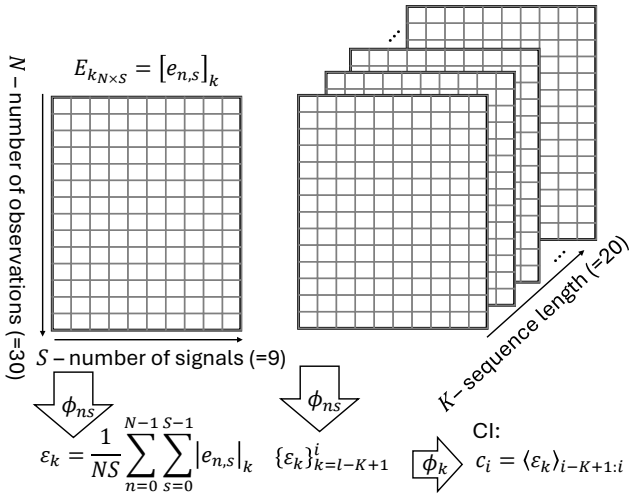


Figure 1. Definition of the reconstruction error and the Mean CI, evaluated at point i in time. Typical parameter values for S, N, and K are indicated in the parentheses.

We define a CI as a real-valued function ϕ_k operating on a

reconstruction-error sequence of length K:

$$\begin{aligned} \phi_\theta &: \mathbb{R}^{K\times N\times S} \to \mathbb{R} \\ \phi(E;\theta) &:= \phi_\theta(E) \end{aligned} \tag{2}$$

where K is the sequence length or number of taps and θ represents parameters derived from the baseline errors $\{E_i\}_{i\in I_B}$, such as an empirical probability distribution. For notational convenience, we typically omit the θ.

All CIs of interest in this paper are more structured than this most abstract definition. Specifically, all of our CI functions $\phi(E;\theta)$ can be decomposed into a real-valued function ϕ_{ns} on a single E_i and a real-valued function ϕ_k applied elementwise to a K-sequence of ϕ_{ns} outputs. Symbolically,

$$\begin{aligned} \phi(\{E_k\}_{k=i-K+1}^{i}) = \\ \phi_k(\phi_{ns}(E_{i-K+1}), \phi_{ns}(E_{i-K+2}), ..., \phi_{ns}(E_i)) \end{aligned} \tag{3}$$

Figure 1 illustrates the two-step process applied to computation of the Mean CI, the simplest CI and one we can use as a baseline for comparison. First, ϕ_{ns} compressed a sample reconstruction error matrix E_i to a single point denoted by a black dot - the Mean Absolute Error (MAE). Second, ϕ_k used K dots to compute the Mean CI as a moving average.

$$\begin{aligned} \varepsilon_i = \phi_{ns}\left([e_{n,s}]_i\right) = \mathrm{MAE}\left([e_{n,s}]_i\right) = \frac{1}{NS}\sum_{\substack{0\le s<S \\ 0\le n<N}} |e_{n,s}|_i \\ c_i = \phi_k(\{\varepsilon_k\}_{i-K+1:i}) = \langle\varepsilon_k\rangle_{i-K+1:i} = \frac{1}{K}\sum_{i-K<k\le i} \varepsilon_k \end{aligned} \tag{4}$$

To simplify the notation, in the remainder of the text, we use $\langle\cdot\rangle_K$ instead of $\langle\cdot\rangle_{i-K+1:i}$. These decompositions reveal avenues for experimentation, not all of which are exhausted in this paper. We may consider alternatives to MAE such as Mean Squared Error (MSE). We may consider larger changes to ϕ_{ns} to alter how the per-signal errors interact. And we may consider alternatives to ϕ_k to change how we consider the sequence of errors. In this paper we focus on CIs with ϕ_{ns} = MAE, and ϕ_k as some statistical measurement. See Table 2 for a list and definitions of CIs we investigated.

The normed negative log-likelihood (NNLL) CI employed the estimated probability density function (PDF) $\hat{p}_t$ obtained from the reconstruction error of the training data in the baseline period. The Gaussian kernel density estimator implemented in Scipy (Virtanen et al., 2020), used the samples from the baseline training data to produce this PDF. The motivation for normalizing the negative likelihood instead of taking the traditional sum is to keep the scale of the CI independent of K. This normalization (or any positively oriented affine transformation of a CI) does not affect our subsequent detection procedures or performance evaluations since our detectors merely operate on margins.

Table 2. CI Definitions Given by Functional Decompositions.

CI Name	$\phi_{ns}\,(E_i)$	$\phi_k\left(\{\varepsilon_k\}_{k=i-K+1}^{i}\right)$
Mean	MAE	$\langle\varepsilon_k\rangle_K$
Median (K even)	MAE	$\langle \mathrm{sort}(\{\varepsilon_k\})_{[K/2-1,K/2]}\rangle$
STD	MAE	$\sqrt{\langle(\varepsilon_k - \langle\varepsilon_j\rangle_K)^2\rangle_K}$
Mean + STD	MAE	$\langle\varepsilon_k\rangle_K + \sqrt{\langle(\varepsilon_k - \langle\varepsilon_j\rangle_K)^2\rangle_K}$
Mean - STD	MAE	$\langle\varepsilon_k\rangle_K - \sqrt{\langle(\varepsilon_k - \langle\varepsilon_j\rangle_K)^2\rangle_K}$
NNLL	MAE	$\langle -\log\hat{p}_t\,(\varepsilon_k)\rangle_K$
Kurtosis	MAE	$\frac{\langle(\varepsilon_k-\langle\varepsilon_j\rangle_K)^4\rangle_K}{\langle(\varepsilon_k-\langle\varepsilon_j\rangle_K)^2\rangle_K^2}$

2.2. Detection

2.2.1. Margins

Detection operates on CIs, but for the purpose of anomaly detection the actual CI values are somewhat irrelevant. What matters most is how they compare to the CI values in the baseline period. Let I_B be the index set of the baseline data, and I_F the index set of the fault window. Fix the number of taps K and causally define our CI values c_i, zero-based index $i \ge K-1$ as

$$c_i = \phi\left(\{E_k\}_{k=i-K+1}^{i}\right) \tag{5}$$

We then construct an affine mapping μ to convert the c_i to margins:

$$\begin{aligned} c_m &= \min_{i\in I_B} c_i \\ c_M &= \max_{i\in I_B} c_i \\ m_i := \mu(c_i) &= \frac{c_i - c_M}{c_M - c_m} \end{aligned} \tag{6}$$

This maps the maximum baseline c_i to 0, the minimum baseline c_i to -1, and in general the margin tells you how far outside of the baseline range you are. We choose CI functions so that we expect anomalies to exceed 0% margin, rather than being under the minimum. Using margins instead of particular CI values lets us describe the detection procedures in a manner agnostic to our choice of ϕ.

2.2.2. Detectors

A detection algorithm is a causal system with respect to the sequence of input margins m_i, and returns the position at which time detection occurred. If no detection occurs by the end of the sequence, the detector returns $null$. We also define the detection horizon h as the (positive) amount of time between detection and the fault event, or 0 if no detection occurred.

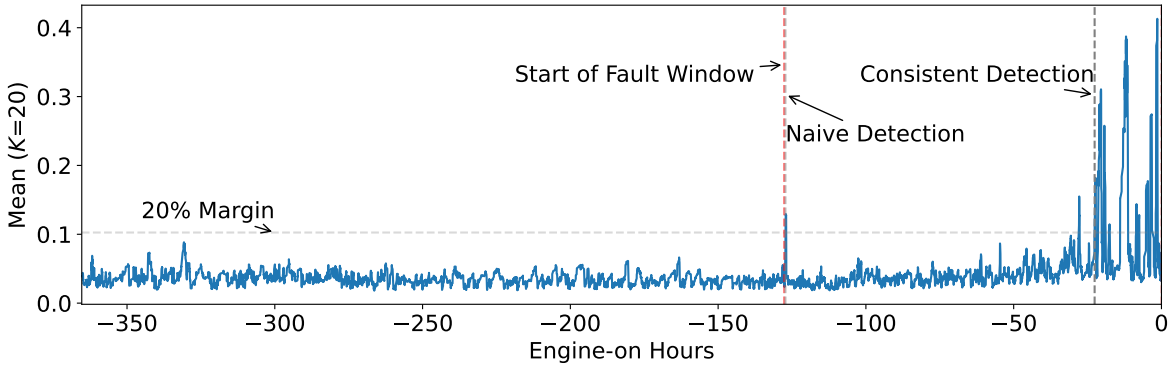


Figure 2. Naïve and Consistent Detection on E04.
$\alpha = 1$ hr, $\beta = 50$ hr

We considered two detection procedures: "naïve" and "consistent", each parameterized by a sensitivity margin $M \geq 0$. Naïve detection simply identifies the first position in the fault window when the margin M is exceeded:

$$\text{detect}_{\text{naïve}}([m_i]; M, I_F) = \min_{i \in I_F}\{i \mid m_i > M\} \tag{7}$$

Consistent detection aims to reduce the effect of isolated outliers and increase our confidence of a true positive. We do this by restricting detection to when the m_i stay above M for a sustained period α, or when we see multiple spikes in m_i above M within some period β. For the α, β conditions to be coherent with each other, β must exceed α and we must also impose the condition that the time difference between the first and last qualifying m_i spike must be at least α. This ensures monotonicity of detection horizon with respect to sensitivity margin; a property that any detection routine should possess. An example is shown in Figure 2 where the naïve detector activates almost immediately, while the consistent detector doesn't activate until the CI spikes become more frequent.

Note that for both naïve and consistent detection, even the margin values don't matter beyond their relationship to M. The only characteristic that influences the detector is $\text{sign}(m_i - M)$. The values of m_i still influence our evaluations in the sense that higher margins mean we're more confident in the detection, but they don't influence *when* the detector activates. This may motivate more sophisticated detection methods that are sensitive to the magnitude of excess margin, but they are beyond the scope of this paper.

2.2.3. Detector-Sensitive Margins

Now that we have defined an alternative procedure to naïve detection, it is fruitful to revisit the margin definitions (Equation 6). We can motivate our original definition for the margin function μ by the property that

$$\text{detect}_{\text{naïve}}([\mu(c_i)]; 0, I_B) = null \tag{8}$$

i.e. we assigned margins so that naïve detection (Equation 7) doesn't detect anything in the baseline if we set the sensitivity to the minimum $M = 0$.

With this perspective it becomes apparent that for more stringent detection algorithms, we can decrease the value where we assigned a margin of 0, thereby increasing sensitivity for free in the fault window. This gives us a recursive definition for a detector-sensitive margin function μ_D, where c_M can be found to arbitrary precision with a binary search:

$$\begin{aligned} c_m &= \min_{i \in I_B} c_i \\ \mu_D(c_i; c_M) &= \frac{c_i - c_M}{c_M - c_m} \\ c_M &= \inf_{x \in \mathbb{R}}\{x \mid \text{detect}_D([\mu_D(c_i; x)]; 0, I_B) = null\} \end{aligned} \tag{9}$$

In words, the only change from the detector-insensitive case is to move the zero-margin point (c_M) down to the minimum value such that the baseline remains undetected. Figure 3 shows an example where E02 did not initially detect with a $K = 20$ NNLL due to a high spike in the baseline, but when using a consistent detector with detector-sensitive margins, a detection does occur.

2.3. CI Performance Evaluations

We sought to determine which condition indicators produce the most convincing detections. We measure performance by considering the trade-off between sensitivity margin M and detection horizon h. If we set a high margin, we may not detect as far in advance (or at all), and if we decrease the margin we may discover an anomaly sooner. Considering h as a function of M, $h(M)$ must be nonincreasing. We refer to $h(M)$ as the detection curve.

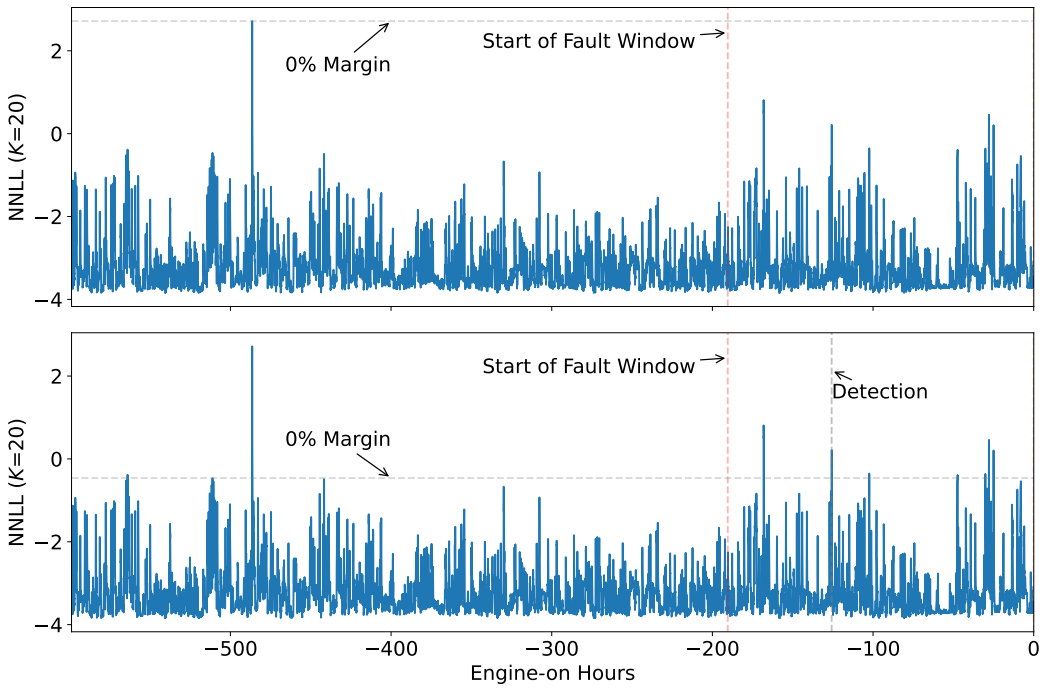


Figure 3. Consistent Detector-Sensitive Margins yielding detection on E02.

Consider the plot of h vs M for E04 with a $K = 20$ Mean CI visualized in Figure 4. The outer corner points of this curve are Pareto optimal with respect to these two metrics. To compress this curve to a single representative metric, we compute the area bounded by the curve and the axes. This can be adjusted to account for a minimum considered margin by only integrating to the right of M_{min}. Given multiple CIs and a fixed detector running on a fixed dataset, we can compare the areas under the detection curves to determine relative CI performance (Figure 5).

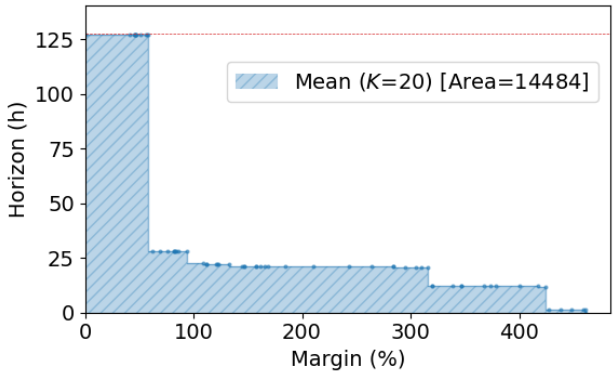


Figure 4. Detection Horizon vs Sensitivity Margin for Mean (K=20) on E04 with Naïve detection.

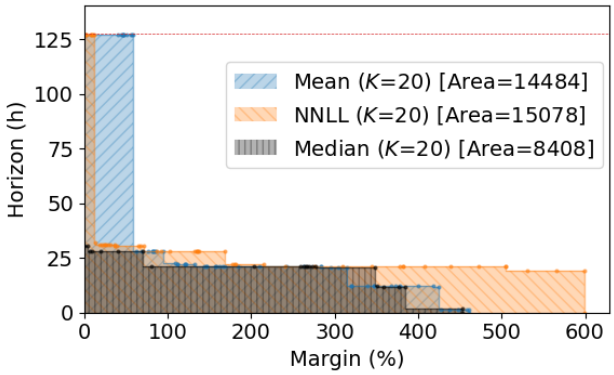


Figure 5. Comparison of 3 CIs on E04 with Naïve detection.

2.3.1. Performance Normalization

The area under the detection curve is not an easily interpreted figure. It has unusual units (% margin $\times$ engine-on-hours), and the expected scale of the number changes from vehicle to vehicle due to variations in the size of the fault window, and the unknown inherent detectability of the vehicles' data. Furthermore, margin has no upper bound and this area gives undesirably high weight to extreme margins. To give a more interpretable measurement, normalize h and m. h normalization is straightforward; since it already falls in a bounded range we can affinely map it to $[0, 1]$. This h mapping is pa-

rameterized by the minimum and maximum horizons h_{min}, h_{max} we seek to assign positive performances to. To normalize the unbounded m, we apply an $arctan$ mapping into $[0, 1]$. This m mapping is parameterized by the minimum margin we seek to assign positive performance to M_{min}, and the real-valued ρ representing the saturation point of the $\arctan$. We kept ρ fixed at $\rho = 500\%$. This normalization makes it easier to compare CIs on a particular vehicle, but should not be used to make comparisons across vehicles.

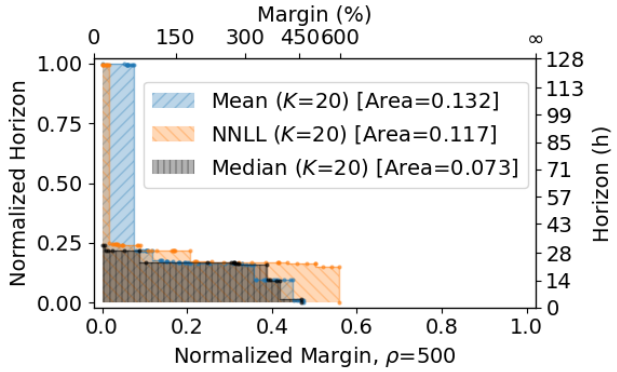
$$\begin{aligned} \hat{h} &= \frac{h - h_{min}}{h_{max} - h_{min}} \\ \hat{m}_\rho &= \frac{2}{\pi} \arctan\left(\frac{m - M_{min}}{\rho - M_{min}}\right) \end{aligned} \tag{10}$$

After transforming to normalized space $[0, 1] \times [0, 1]$, we measure the area under the normalized curve A_{norm} to give a performance metric. This normalized area A_{norm} can be considered as a percentage of optimal performance, where the (unattainable) optimum would be an infinite margin as far in advance as possible (Figure 6). Table 3 shows the normalized performance for each CI with K fixed at 20 on E04's idle data. The Mean CI was the only CI under consideration in our prior work (Kohrt et al., 2024) and serves as the benchmark for the performance of other CIs. By this area metric, NNLL performed the best, narrowly edging out STD. However, this ranking of CIs is not consistent across all vehicles and all K. Table 4 shows the top 3 CIs per vehicle with K options of $\{20, 50, 100, 200, 500\}$, all evaluated on idle data. This table shows that there is no globally optimal CI, though NNLL tends to be a high performer at various K. In addition to the CI rankings, the presence of any positive performance on E02 and E05 is an improvement, as no convincing detection was found on either vehicle in our original postprocessing routine.

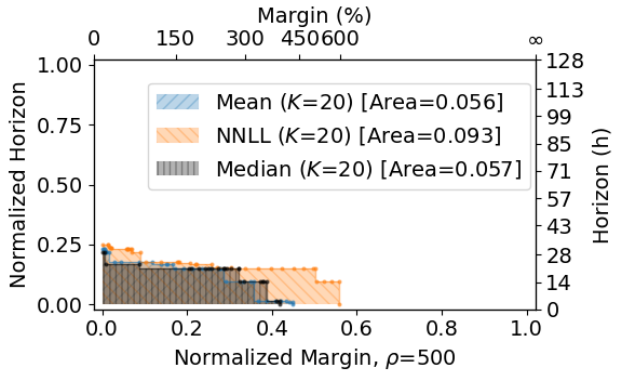
Table 3. CI Performances on E04 with K=20.

CI Name	A_{norm}	Horizon (h)	Margin (%)
Mean	0.076	29.6	32
Median	0.081	32.7	28
STD	0.095	31.7	43
Mean + STD	0.086	30.7	55
Mean - STD	0.052	21.3	27
NNLL	**0.098**	**29.6**	**67**
Kurtosis	0.002	32.1	4

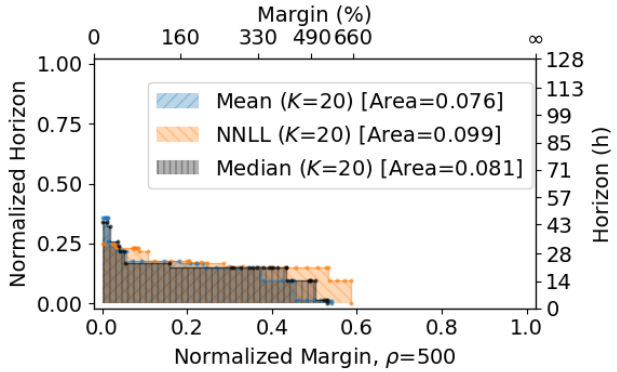
While some credit for new detections goes to the new CIs, it was the introduction of detector-sensitive (D-S) margins coupled with the consistent detector that was the most impactful. It is easy to see mathematically that with a fixed detector, detector-sensitive margins cannot reduce performance as we measure it; though it is not a given that they can yield new detections. But in our case, by using consistent detection and D-S margins, the brief and unconvincing anomalies in the baseline are filtered out, and the more frequent and convincing CI spikes in the fault window are emphasized.



(a)



(b)



(c)

Figure 6. Detection Curves on E04 in Normalized Space. (a) Naïve, (b) Consistent (without D-S margins), (c) Consistent (with D-S margins).

Table 4. Top 3 CIs per vehicle operating on idle-only data, ranked by normalized area of the Consistent detection curve. Vehicles marked with an asterisk (*) are those which did not have detection under Mean in our original postprocessing procedure which used $K \leq 100$. Daggers (†) indicate cases where a detection technically occurred but we deemed it to be unconvincing.

ID	CI Name	K	A_{norm}	Horizon (h)	Margin (%)
E01	NNLL	20	0.052	29.7	24
	NNLL	500	0.043	19.8	53
	NNLL	50	0.030	19.6	36
E02*	NNLL	20	0.017	125.7	20
	NNLL†	50	0.006	168.0	5
	-	-	-	-	-
E03	STD	500	0.038	25.9	53
	STD	200	0.027	27.1	37
	Mean-STD	20	0.027	26.9	41
E04	Kurtosis	500	0.651	121.7	861
	STD	100	0.348	19.6	422
	STD	200	0.302	19.3	487
E05*	Mean	500	0.005	44.8	11
	NNLL	500	0.004	44.8	10
	NNLL	200	0.004	71.8	5
E06	NNLL	500	0.306	207.6	236
	Median	20	0.256	211.2	162
	Median	50	0.248	209.7	173
E07*	-	-	-	-	-
	-	-	-	-	-
	-	-	-	-	-
E08*	STD†	500	0.005	12.4	6
	STD†	100	0.001	16.6	1
	STD†	200	0.001	16.2	1

3. FAULT CLASSIFICATION

In addition to detecting anomalies, we wanted to diagnose some aspects of the fault. This data-driven diagnostics trained and evaluated classifiers on the autoencoder reconstruction error associated with anomalies. Sections 3.1-3.2 overview a classification approach with initial promise, and Section 3.3 describes some robustness checks with concerning results.

3.1. Dataset

Table 5 lists the seven failure modes and their descriptions used in the study, as well as the distribution of twenty-two vehicles over the seven failure modes.

Vehicle IDs starting with a "T" had a transmission-related fault, and vehicles starting with "E" had an engine-related fault. These broad fault categories have slightly different signal sets associated with them, so for this combined experiment the union of these signal sets was used for autoencoder training and subsequent error classification. Additionally, for engine-only models we choose between idle and driving data and some signals are only considered for a particular operation mode. Since all transmission faults are expected to be found during driving conditions, driving data is used for this combined experiment. Table 6 indicates which signals are present and which fault categories they are associated with.

Table 5. Failure mode definitions.

FM	FM Description	Count	Vehicle IDs
TL	transmiss. leak	7	T04, T08, T09, T10, T11, T12, T13
FID	fuel injector damage	5	E04, E05, E13, E20, E21
CL	coolant low	3	E06, E07, E08
TI	transmiss. inoperable	2	T06, T18
TC	transmiss. cooler	2	T05, T07
CF	cooling fault	2	E01, E02
TF	thermostat failure	1	E03

Table 6. Signals included in autoencoder models. Asterisks indicate virtual (computed) signals.

Signal Name	Engine	Transmission
Vehicle Speed	Driving Only	No
Engine Speed	Yes	Yes
Torque	Yes	Yes
Shaft Speed	No	Yes
Engine Coolant Temp.	Yes	Yes
Engine Coolant Temp. Gradient*	Yes	No
Engine Power*	No	Yes
Transmission Temp.	No	Yes
Transmission Temp. Gradient*	No	Yes

To maintain the integrity of our model evaluation, we ensured that data from any individual vehicle was used either exclusively for training or exclusively for validation, never both. This separation required a minimum of two vehicles to develop reliable models. However, as shown in Table 5, the thermostat failure (TF) category only had a single vehicle of that type, making it unsuitable for our modeling approach.

Because some failure modes required certain operating conditions to be observed, only the data corresponding to the anomalies were qualified for model training and validation. However, to maximize the size of the dataset, we reduced the detection threshold margin to zero, as illustrated in Figure 7.

For reliable anomaly detection, our classification models required multiple consecutive data points from a single vehicle to establish confidence in its predictions. We set this threshold at $m = 200$ samples, a choice we justify in our later discussion of model performance. Among all failure modes, only three had sufficient data from multiple vehicles to support model development: coolant low (CL), fuel-injector damage (FID), and transmission leak (TL). The remaining failure modes – TF, cooling fault (CF), transmission inoperable (TI), and transmission cooler (TC) – lacked adequate vehicle representation for meaningful model development.

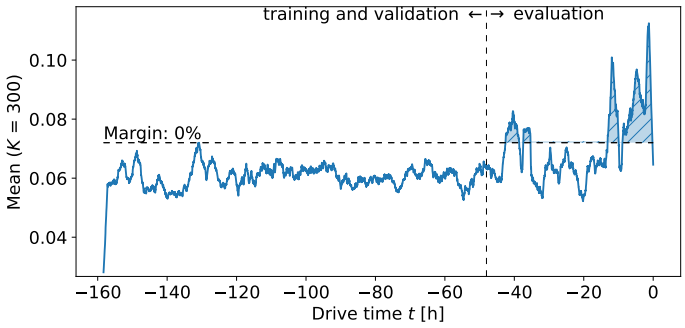


Figure 7. Example of a valid data range used for classification (Veh. ID = E04).

Table 7. Vehicles with $m \geq 200$ samples above the margin.

FM	Vehicle ID	data
CL	E06	2,998
CL	E08	740
FID	E04	2,311
FID	E21	435
TC	T07	775
TF	E03	1,026
TI	T18	1,861
TL	T04	754
TL	T09	13,190
TL	T10	773
TL	T11	1,247
TL	T12	1,375
TL	T13	3,454

3.2. Classifier Construction and Results

Table 8 shows the vehicle FM representation for model development based on using a similar number of data points for training and a similar number for validation without class balancing.

Table 8. Vehicle selection for classification.

	Training		Validation	
FM	Vehicle ID	Samples	Vehicle ID	Samples
CL	E06	2,998	E08	740
FID	E04	2,311	E21	435
TL	T13	3,454	T04	754

The classification model was the random forest classifier implemented in Scikit-learn (Pedregosa et al., 2011), with number of estimators set to 30 and maximum depth set to 10. The model operated on randomly permuted samples. A single-point classification results are shown in Figure 8. The model's effectiveness was assessed by examining the confusion matrix: when the diagonal element is the highest value in its column, classification was reinforced by combining classifier outputs for multiple sample inputs.

For example, Figure 9 shows the confusion matrix when 200 consecutive points merged using a mode (majority-based) filter. Note that the validation confusion matrix, in this FM ordering, has a block-diagonal structure, where engine-related FMs, CL and FID, are mutually less distinguishable compared to TL, i.e., FID and CL form a diagnostic ambiguity group relative to TL. The top plot in Figure 10 shows how the mode filter operates on consecutive samples. There is a

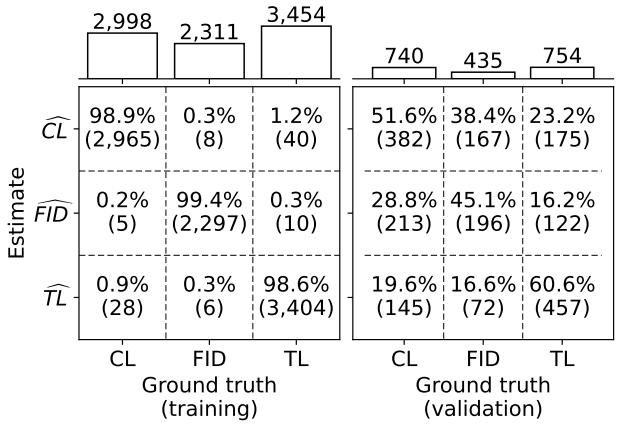


Figure 8. Single-point classification.

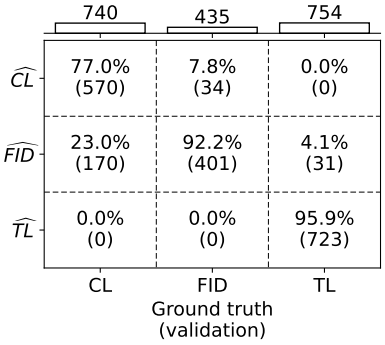


Figure 9. Multi-point classification.

physics-based explanation: this ambiguity is due to the effect that fuel-injector faults can have on the actual thermal load of the engine relative to the anticipated thermal load based on the engine's operation with healthy fuel injectors. This effect was seen in the temperature of the exhaust gas in a diesel engine as increasingly severe injector faults were tested (Thurston, Sullivan, & McConky, 2023).

Similarly, a coolant leak that has resulted in a low coolant level would have an effect on the ability of the engine to transfer heat to its radiator, affecting the engine's coolant temperature relative to the behavior of a healthy vehicle. Whereas a transmission oil leak resulting in low transmission oil levels would have an effect on transmission oil temperature. While the transmission oil is cooled by the engines coolant, a transmission related fault would still have a far lesser effect on

engine coolant temperature then a direct engine related fault, hence why the transmission related fault is more distinguishable.

In addition to mode filtering, we used Bayesian fusion and the Dirichlet distribution as the conjugate prior to the multinomial, as illustrated in the bottom Figure 10 as a function of samples and in the bottom Figure 11 as it evolves in time, relative to the CI and anomaly detection. After observing only a few samples, the model assigned about 60% probability to the correct failure mode but did not improve further with more data.

3.3. Classifier Robustness

Only three of the failure modes discussed above had representation in more than two vehicles, and we tested diagnostics for only one partition of vehicles. Given that, the results required additional scrutiny. Specifically, although it was reasonable to expect a classifier to perform best when the classes were naturally balanced, it was also necessary to investigate the classifier's performance on different vehicles to test the robustness of the approach. The robustness investigation focused on binary classification between the TL and CL failure modes because only two vehicles had FID faults. Our choices of CL vehicles were still limited, but we could at least validate our TL/CL classifier with data from other TL vehicles. Unfortunately, only half of these alternate TL vehicles came away with positive classification performance.

To make the situation worse, we found that the classification was sensitive to small perturbations in autoencoder reconstruction error. The autoencoder perturbations were induced by extending the training by a few epochs (resuming exactly from the optimizer state at which they were left in the original training). Specifically, we took each autoencoder and independently trained for 5, 10, and 20 more epochs to produce a set of four autoencoders for each vehicle. If our classifiers were learning features intrinsic to the fault types, they should still validate well on reconstruction error from slightly different autoencoder models of the same vehicles. However, we found that this was not the case. Multiple repeated machine learning experiments resulted in at least one of the four perturbed autoencoders generating reconstruction error that was not validated successfully. The model was not so fragile that any perturbation would cause it to fail, but some seemingly innocuous perturbations did.

We attempted to remove the sensitivity to the autoencoder state using two approaches. The first, based on augmented datasets that contain multiple autoencoder states, was not successful. The second approach started by normalizing reconstruction error vector to unit length. The motivation was to reinforce the idea that different FMs have unique signatures along the reconstruction error vector (which is formed by concatenating error vectors of individual signals). Thus, the classifiers were forced to focus on the information associated with *directions* of these unit vectors while ignoring their original magnitude. The normalization, combined with a dataset augmentation that exploited the observation that autoencoder perturbations tend to move the error vectors by relatively small angles, made some classifiers robust to the autoencoder perturbations. Unfortunately, the classifiers still failed to consistently generalize to other validation vehicles.

These robustness tests strongly suggests that, at least for our data and autoencoder models, classifiers based on reconstruction error are fragile.

4. CONCLUSION

We investigated selected paths to enrich the postprocessing of autoencoder error to improve threshold-based anomaly detection by increasing the related detection horizon and sensitivity margin. Because of the trade-off between these two metrics, the evaluation metric was the area under the curve of the Pareto front with horizon-margin coordinates. Normalizing the coordinates of the horizon and the margin improves the metric interpretation. Using the normalized area under the Pareto curve, the best performing CI was NNLL (though it was not entirely dominant, see Table 3).

In addition to the strict threshold-based anomaly detector (that is, the naïve detector), we investigated a consistent detector that considered the persistence of the threshold exceedance using the duration and recurrence of threshold crossings. The consistent detector was not only less sensitive to outliers, but also enabled detection in some models where the naïve method failed (see Figure 3).

The full reconstruction error showed promise as a set of feature vectors to classify anomalies. To maximize the size of the dataset, the classifier used the reconstruction error associated with crossing the zero-margin threshold. Consecutive classifications were combined using two methods: a simple mode filter and Bayesian fusion. However, the anomaly classifiers tended to be fragile in their generalization ability, a cause for concern and further investigation. We plan to assess this further by examining the performance of our techniques on higher-resolution data with crisper anomalies, such as distinguishing surface damage from fatigue cracking in gearbox vibration data.

FUNDING

This research was sponsored by the Department of the Navy, Office of Naval Research under ONR award number N00014-19-1-2600.

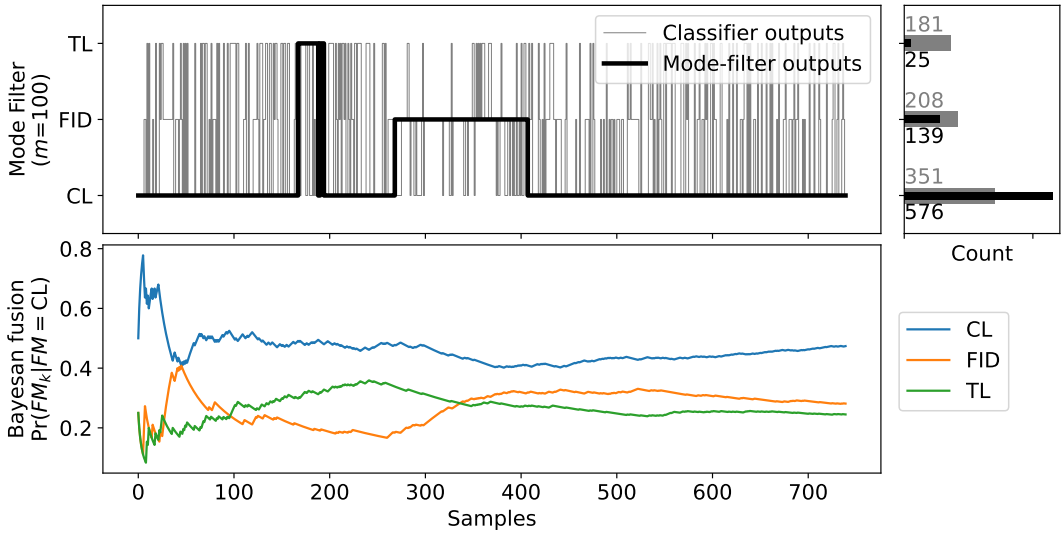


Figure 10. Example of mode filtering (top) and Bayesian fusion (bottom) for the ground truth associated with lowest-performing class, i.e., CL.

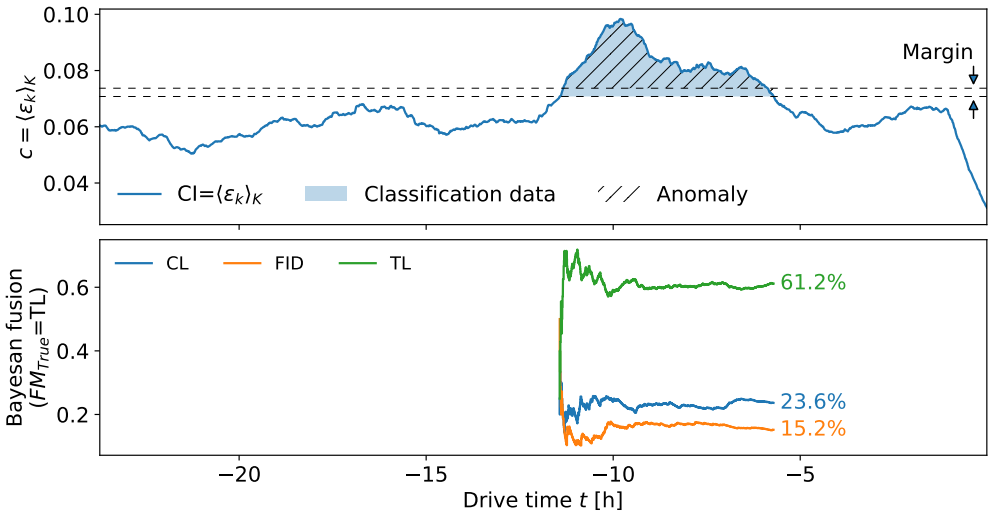


Figure 11. Time evolution of the Mean CI $c = \langle \varepsilon_i \rangle_K$ with associated diagnostics.

DISCLAIMER

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Office of Naval Research.

REFERENCES

Adkisson, M., Kimmell, J. C., Gupta, M., & Abdelsalam, M. (2021). Autoencoder-based anomaly detection in smart farming ecosystem. *Proceedings - 2021 IEEE International Conference on Big Data, Big Data 2021*, 3390–3399.

Ahmad, S., Styp-Rekowski, K., Nedelkoski, S., & Kao, O. (2021). *Autoencoder-based condition monitoring and anomaly detection method for rotating machines.* arXiv.

Bezak, N., Brilly, M., & Šraj, M. (2014). Comparison between the peaks-over-threshold method and the annual maximum method for flood frequency analysis. *Hydrological Sciences Journal*, *59*(5), 959–977.

Connelly, A. C., Zaidi, S. A. R., & McLernon, D. (2023, April). Autoencoder and incremental clustering-enabled anomaly detection. *Electronics*, *12*(9), 1970.

de Pater, I., & Mitici, M. (2023). Developing health indicators and rul prognostics for systems with few failure instances and varying operating conditions using a lstm

autoencoder. *Engineering Applications of Artificial Intelligence*, *117*, 105582.

Deng, Z., Wang, Z., Tang, Z., Huang, K., & Zhu, H. (2021). A deep transfer learning method based on stacked autoencoder for cross-domain fault diagnosis. *Applied Mathematics and Computation*, *408*, 126318.

Hsu, C.-C., Frusque, G., & Fink, O. (2023, October). Comparison of residual-based methods on fault detection. *Annual Conference of the PHM Society*, *15*(1).

Kohrt, E., Moon, M., Sullivan, M., Das, S., Thurston, M., & Nenadic, N. G. (2024). Data-driven detection of engine faults in infrequently-driven ground vehicles. In *Annual conference of the phm society* (Vol. 16).

Kreuzer, M., & Kellermann, W. (2023, June). 1-d residual convolutional neural network coupled with data augmentation and regularization for the icphm 2023 data challenge. In *2023 ieee international conference on prognostics and health management (icphm).* IEEE.

Krishnan, R. S., Gopikumar, S., Muthu, A. E., Raj, J. R. F., Kumari, D. A., & Malar, P. S. R. (2024, June). Next-gen manhole monitoring: Autoencoder-assisted anomaly detection. In *2024 3rd international conference on applied artificial intelligence and computing (icaaic)* (p. 1426–1433). IEEE.

Lachekhab, F., Benzaoui, M., Tadjer, S. A., Bensmaine, A., & Hamma, H. (2024, May). Lstm-autoencoder deep learning model for anomaly detection in electric motor. *Energies*, *17*(10), 2340.

Lagazo, D., de Vera, J., Coronel, A., Jimenez, J., & Gatmaitan, E. (2021, June). Condition-based monitoring and anomaly detection of industrial equipment using autoencoder. In *2021 international conference on artificial intelligence and computer science technology (icaicst)* (p. 146–151). IEEE.

Liang, Q., Knutsen, K. E., Vanem, E., Zhang, H., & Æsøy, V. (2023, September). Unsupervised anomaly detection in marine diesel engines using transformer neural networks and residual analysis. *PHM Society Asia-Pacific Conference*, *4*(1).

Mallak, A., & Fathi, M. (2021, January). Sensor and component fault detection and diagnosis for hydraulic machinery integrating lstm autoencoder detector and diagnostic classifiers. *Sensors*, *21*(2), 433.

Michau, G., Hu, Y., Palmé, T., & Fink, O. (2019, August). Feature learning for fault detection in high-dimensional condition monitoring signals. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, *234*(1), 104–115.

Park, P., Marco, P. D., Shin, H., & Bang, J. (2019, October). Fault detection and diagnosis using combined autoencoder and long short-term memory network. *Sensors*, *19*(21), 4612.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... others (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, *12*(Oct), 2825–2830.

Peixoto, J., Sousa, J., Carvalho, R., Soares, M., Cardoso, R., & Reis, A. (2023, August). Anomaly detection with a lstm autoencoder using influxdb. In *Flexible automation and intelligent manufacturing: Establishing bridges for more sustainable manufacturing systems* (p. 69–76). Springer Nature Switzerland.

Reddy, K. K., Sarkar, S., Venugopalan, V., & Giering, M. (2016, oct). Anomaly Detection and Fault Disambiguation in Large Flight Data: A Multi-modal Deep Autoencoder Approach. *Annual Conference of the PHM Society*, *8*(1), 192–199.

Ryu, G., & Seong, N. (2023, October). Anomaly detection and fault classification in multivariate time series using multimodal deep models. *Annual Conference of the PHM Society*, *15*(1).

Ryu, S., Jeon, B., Seo, H., Lee, M., Shin, J.-W., & Yu, Y. (2023, February). Development of deep autoencoder-based anomaly detection system for hanaro. *Nuclear Engineering and Technology*, *55*(2), 475–483.

Shao, H., Jiang, H., Wang, F., & Zhao, H. (2017, March). An enhancement deep feature fusion method for rotating machinery fault diagnosis. *Knowledge-Based Systems*, *119*, 200–220.

Shen, H., Wang, X., Fu, L., & Xiong, J. (2023, June). Gear fault diagnosis based on short-time fourier transform and deep residual network under multiple operation conditions. In *2023 ieee international conference on prognostics and health management (icphm)* (p. 166–171). IEEE.

Shvetsova, N., Bakker, B., Fedulova, I., Schulz, H., & Dylov, D. V. (2021). Anomaly detection in medical imaging with deep perceptual autoencoders. *IEEE Access*, *9*, 118571–118583.

Siffer, A., Fouque, P. A., Termier, A., & Largouet, C. (2017, aug). Anomaly detection in streams with extreme value theory. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1067–1075.

Thill, M., Konen, W., Wang, H., & Bäck, T. (2021, nov). Temporal convolutional autoencoder for unsupervised anomaly detection in time series. *Applied Soft Computing*, *112*, 107751.

Thurston, M. G., Sullivan, M. R., & McConky, S. P. (2023). Exhaust-gas temperature model and prognostic feature for diesel engines. *Applied Thermal Engineering*, *229*, 120578.

Tian, R., Liboni, L., & Capretz, M. (2022, November). Anomaly detection with convolutional autoencoder for predictive maintenance. In *2022 9th international conference on soft computing & machine intelligence (iscmi)* (p. 241–245). IEEE.

Torabi, H., Mirtaheri, S. L., & Greco, S. (2023, January).

Practical autoencoder based anomaly detection by using vector reconstruction error. *Cybersecurity*, *6*(1).

Tuli, S., Casale, G., & Jennings, N. R. (2022, February). Tranad: deep transformer networks for anomaly detection in multivariate time series data. *Proceedings of the VLDB Endowment*, *15*(6), 1201–1214.

Tziolas, T., Papageorgiou, K., Theodosiou, T., Papageorgiou, E., Mastos, T., & Papadopoulos, A. (2022, June). Autoencoders for anomaly detection in an industrial multivariate time series dataset. In *The 8th international conference on time series and forecasting* (p. 23). MDPI.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., … SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272.

Vuong, N. K., Giduthuri, S. B., Lim, G. L., Tan, T., & Ramasamy, S. (2024). Anomaly detection and breakdown diagnosis for condition monitoring of marine engines. *Proceedings - 2024 IEEE Conference on Artificial Intelligence, CAI 2024*, 200–205.

Yan, W., Guo, P., gong, L., & Li, Z. (2016). Nonlinear and robust statistical process monitoring based on variant autoencoders. *Chemometrics and Intelligent Laboratory Systems*, *158*, 31-40.

Zhang, C., Hu, D., & Yang, T. (2022, June). Anomaly detection and diagnosis for wind turbines using long short-term memory-based stacked denoising autoencoders and xgboost. *Reliability Engineering & System Safety*, *222*, 108445.

BIOGRAPHIES

Matthew E. Moon received his B.S. in Computer Engineering and Mathematics from Rose-Hulman Institute of Technology (Terre Haute, IN, USA) in 2018 and his MEng in Electrical Engineering from Rice University (Houston, TX, USA) in 2019. He joined the Golisano Institute for Sustainability (GIS) at Rochester Institute of Technology in 2023 where he is currently working as a Data Science Engineer.

Ethan A. Kohrt received his B.S. in Computer Science from Furman University (Greenville, SC, USA) in 2021 and his M.S. in Computer Science from Northwestern University (Evanston, IL, USA) in 2022. In 2023 he joined the Golisano Institute for Sustainability where he is currently a Data Science Engineer.

Michael G. Thurston received his B.S. and M.S. in Mechanical Engineering from Rochester Institute of Technology (Rochester, NY, USA) in 1988, and his Ph.D. in Mechanical and Aerospace Engineering from the University of Buffalo (Buffalo, NY, USA) in 1998. He is the Technical Director and Research Associate Professor at the Center of Integrated Manufacturing Studies at Rochester Institute of Technology. He formerly held positions in air conditioning system development at General Motor and Delphi, and as a Researcher at the Applied Research Laboratory at Penn State University. He holds 7 patents in the areas of air conditioning and asset health monitoring. His research interests include: sustainable design and production, condition based maintenance and prognostics, and asset health management. He is a member of the Society of Automotive Engineers, and was awarded the Boss Kettering Award for product innovation by Delphi.

Nenad G. Nenadic received his B.S. in Electrical Engineering from University of Novi Sad (Novi Sad, Serbia) in 1996 and his MS and Ph.D. in Electrical and Computer Engineering from University of Rochester (Rochester, NY, USA) in 1998 and 2001, respectively. He joined Kionix Inc. in 2001, where he worked on development of microelectromechanical inertial sensors. Since 2005, he has been at Rochester Institute of Technology, where he is currently a Research Associate Professor. His research interest include design, analysis, and monitoring of electromechanical devices and systems. He co-authored a textbook *Electromechanics and MEMS* and is a member of IEEE.