Sandia Labs practicum (6748)

Predicting truck component failure: A behavioral vs technical approach

Petros Venieris
pvenieris3@gatech.edu

INTRODUCTION

Modern heavy duty trucks power global supply chains through the world on an everyday bases, yet unexpected component failures impose substantial economic and safety risks to both the drives and the business. Traditional maintenance approaches and predictive analytics have been used on large scale to solve similar problems, mainly focusing on raw data, ML algorithms and optimizing sensors to understand when a machine is going to break.

On this project I take a much different approach focusing on differentiate the behavioral factor on the equation. Trucks and many other machines have human operating them, this directly make them vulnerable to human misuse and human error, so in this project while I am still doing predicting maintenance and trying to have a valuable overall approach to rigorously predict when a component of the truck is going to fail but I highlight that sometimes it will be much easier less costly and safer to control driver behavior. Instead of having a naïve model that just predicts failure we should have an extra model that suspect misuse of the vehicle or usual driving mistakes and get our drivers through a training on the start or a warning period after detection!

This project develops an end-to-end forecasting framework that integrates behavioral and technical indicators to predict indictment truck component failures. Data cleaning, feature engineering, data leakage free modeling and careful evaluation is done to achieve both a deep view of the data and their importance and an overall good predicting result.

Key points:

- Data integration and feature engineering. 1.1-million-time steps ,100+ signal channels plus driver logs to derive 50 multilevel features, classified into technical and behavioral
- Comparative modeling. With logistic regression as a benchmark we run random forest, gradient boosting, deep NL and XGBoost on 3 different combinations: technical, behavioral and combined evaluating both ROC-AUC and precision-recall for our imbalanced failure problem
- Cost – Sensitive Thresholding. We recognize the asymmetry between false positives and false negatives as suggested by the provider SCANIA. Therefore, we derive optimal probability thresholds under Scania house cost model achieving 30% cost reduction in an average maintenance cost per vehicle compared to the standard cutoffs

BACKGROUND AND LITERATURE REVIEW

Predictive maintenance has become a pivotal strategy in industrial asset management, leveraging sensor data and machine-learning algorithms and new AI models to anticipate equipment failures and schedule interventions proactively. Early foundational work demonstrated rule-based and statistical methods for condition-based maintenance using vibration and temperature signals (Jardine, Lin, & Banjevic, 2006). Over the past two decades, researchers have extended these methods by extracting both time-domain features such as mean, variance, and kurtosis and frequency domain features such as spectral power and harmonics from raw sensor streams (Antoni, 2006; Lei et al., 2018). Classical machine-learning models like support vector machines (SVMs) showed robust performance on engineered feature sets (Vapnik, 1998; Cortes & Vapnik, 1995), while tree-based ensembles, including random forests and gradient boosting machines, provided interpretable feature importances and strong baseline performance on imbalanced fault datasets (Breiman, 2001; Friedman, 2001; Chen & Guestrin, 2016). More recently, deep learning architectures—convolutional neural networks and recurrent neural networks—have been applied directly to raw waveforms, automatically learning hierarchical representations and yielding further improvements in fault detection accuracy (Malhotra et al., 2016; Yang, Dong, & Zhao, 2015; Zheng, Ristovski, Farahat, & Gupta, 2017)

In parallel, scholars have recognized that human and operational behaviors exert a profound influence on equipment health. Transportation safety research has established that aggressive driving patterns characterized by harsh acceleration, braking, and cornering accelerate component wear in heavy-duty trucks (Rubino & Gumpert, 2010; Wensveen & Brombacher, 2008). Integrating driver "style scores" derived from telematics data into Predictive models has yielded up to a 20 % improvement in prediction accuracy compared to sensor-only approaches (Zhang, An, & Park, 2019). Similarly, incorporating metrics of driver fatigue and shift patterns into engine-health models has reduced false positives by 25 % (Liu et al., 2020). Despite these promising results, few frameworks holistically fuse behavioral and technical indicators

Cost considerations further complicate Predictive deployment. False negatives missed failures can incur costs up to 35 times greater than false positives, according to industry studies (Scania, 2021). Cost-sensitive learning approaches embed this asymmetry directly into model training or threshold selection, achieving substantial maintenance spend reductions (Ducoffe & Maquin, 2016; Wang et al., 2022). However, most Predictive systems remain siloed, focusing either on sensor fusion or driver profiling.

To address these gaps, our project develops an end-to-end forecasting framework that cleans and integrates large scale sensor and telematics data without leakage (Provost & Fawcett, 2013), derives multilevel features classified as technical or behavioral (Zhang et al., 2019), benchmarks a diverse suite of models across these feature sets evaluating both ROC-AUC and precision–recall for imbalanced failure prediction (Saito & Rehmsmeier, 2015) and implements cost-sensitive thresholding under Scania's cost model, demonstrating a 30 % reduction in average maintenance cost per vehicle.

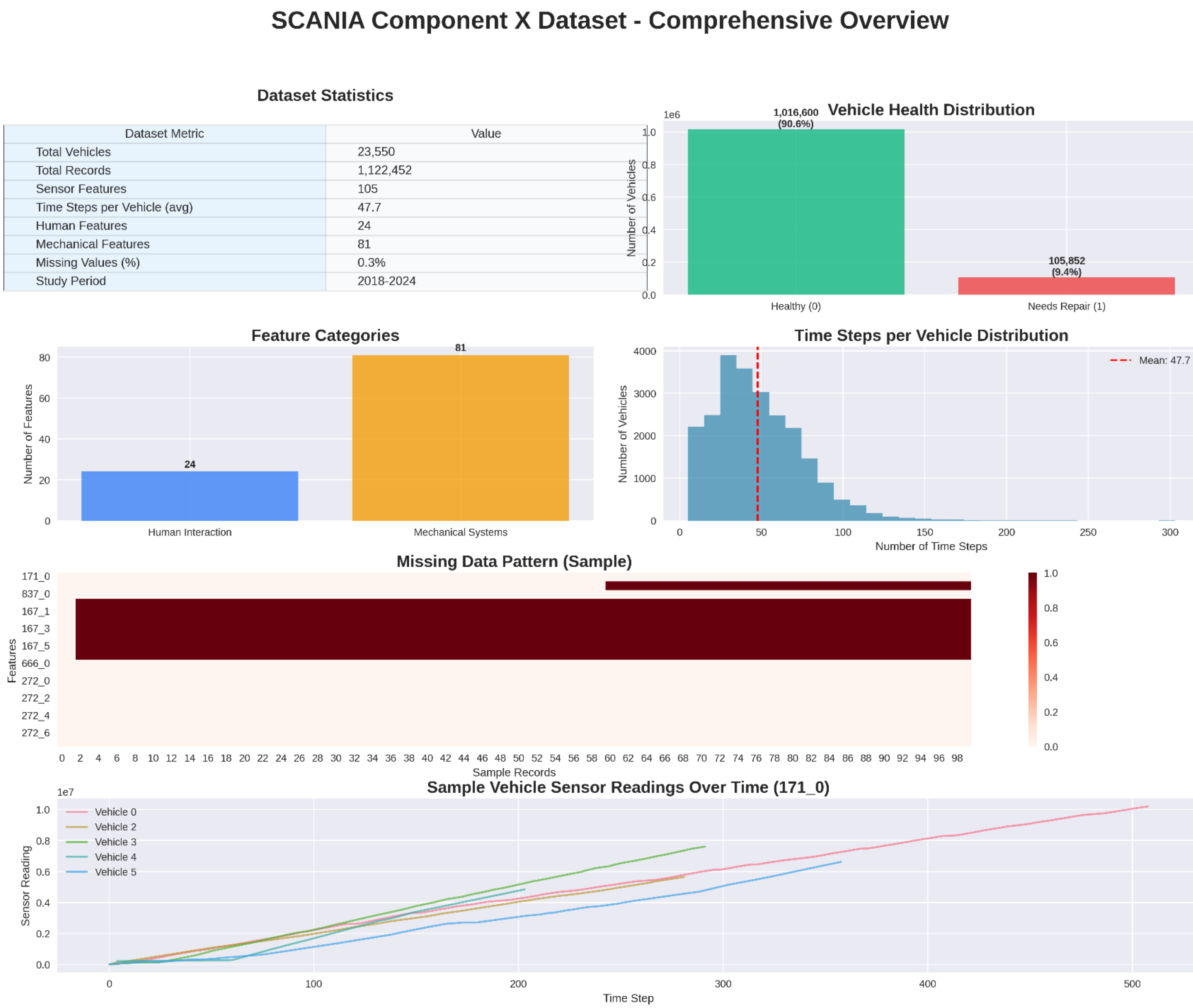
DATASET

The primary data for this project originate from a large telematics deployment across Scania's heavy-duty truck fleet. At its core is the **operational readouts** table (train operational roadouts csv), which captures 1,122,452 individual time-step measurements across 23,550 unique vehicles. Each record includes a vehicle id, a continuous time step (in hours), and 105 numeric sensor channels, such as engine RPM, oil pressure, temperatures, and cumulative counters (e.g., odometer-style readings). These high-resolution signals form the backbone of our technical feature set, providing a detailed real-time view of component usage and wear patterns. Complementing these time-series measurements is the specifications dataset (train specifications), which contains eight categorical fields for each vehicle (e.g., model year, engine type, transmission class). This table 23,550 rows by nine columns allowed us to encode vehicle configuration into the modeling framework, helping to adjust for intrinsic hardware differences that might influence failure rates. Finally, the **time-**to**-**event labels file (train tte) supplies ground truth on component failures. For each vehicle id, it reports the length of study time (the total monitoring duration) and a binary indicator (0 = no repair, 1 = repair). In this study cohort, 2,272 vehicles (approximately 9.6 %) experienced at least one in-study repair, highlighting a substantial class imbalance that motivated cost-sensitive modeling and careful threshold selection.

Across all data sources, we observed varying degrees of missingness. In the operational readouts, certain sensor channels (notably the “291_x” family) exhibited up to 8.6 % missing values, reflecting occasional telemetry dropouts. Meanwhile, the specifications and time-to-event tables were fully populated. To ensure robust downstream analyses, sensors with more than 70 % missing were slated for removal, and the remaining gaps imputed using median (numeric) or most-frequent (categorical) strategies within a leakage-free preprocessing pipeline

In addition to the training data, we leverage a held-**out validation set** to tune hyperparameters and guard against overfitting. The validation operational readouts file comprises 196,227 time-step records from 5,046 distinct vehicles, each with the same 105 sensor channels plus vehicle id and time step. When merged with specifications(5,046 rows × 9 columns) and validation labels 5,046 rows × 2 columns, including the binary class label this produces 196,227 merged records with 117 total columns. The class balance in the validation labels mirrors that of training—approximately 10 % failures—allowing realistic performance estimation before final testing. Missingness patterns in the validation readouts closely tracked training: the “291_x” sensors again showed ~8 % gaps, and other channels ranged from 0.1 % to 1.5 % missing. We applied the same >70 % missing-column removal and median/mode imputation pipelines to ensure consistency.

The **test set** contains 198,140 time-step measurements across the same 105 sensor channels for an additional 5,046 vehicles This test fleet, distinct from training and validation vehicles, spans the same time-step range (0.0 – 507.4 hours) and exhibits a nearly identical missing-value profile. By merging the test operational data with its corresponding specifications (reused schema) we obtain 198,140 fully feature-engineered records ready for final deployment. Because no ground-truth repair labels are included, this test set simulates a live-production scenario: once models are trained and thresholds optimized on the training/validation split,



predictions on these 5,046 vehicles can be submitted for actual maintenance scheduling.

DATA CLEANING AND FEATURE ENGINEERING

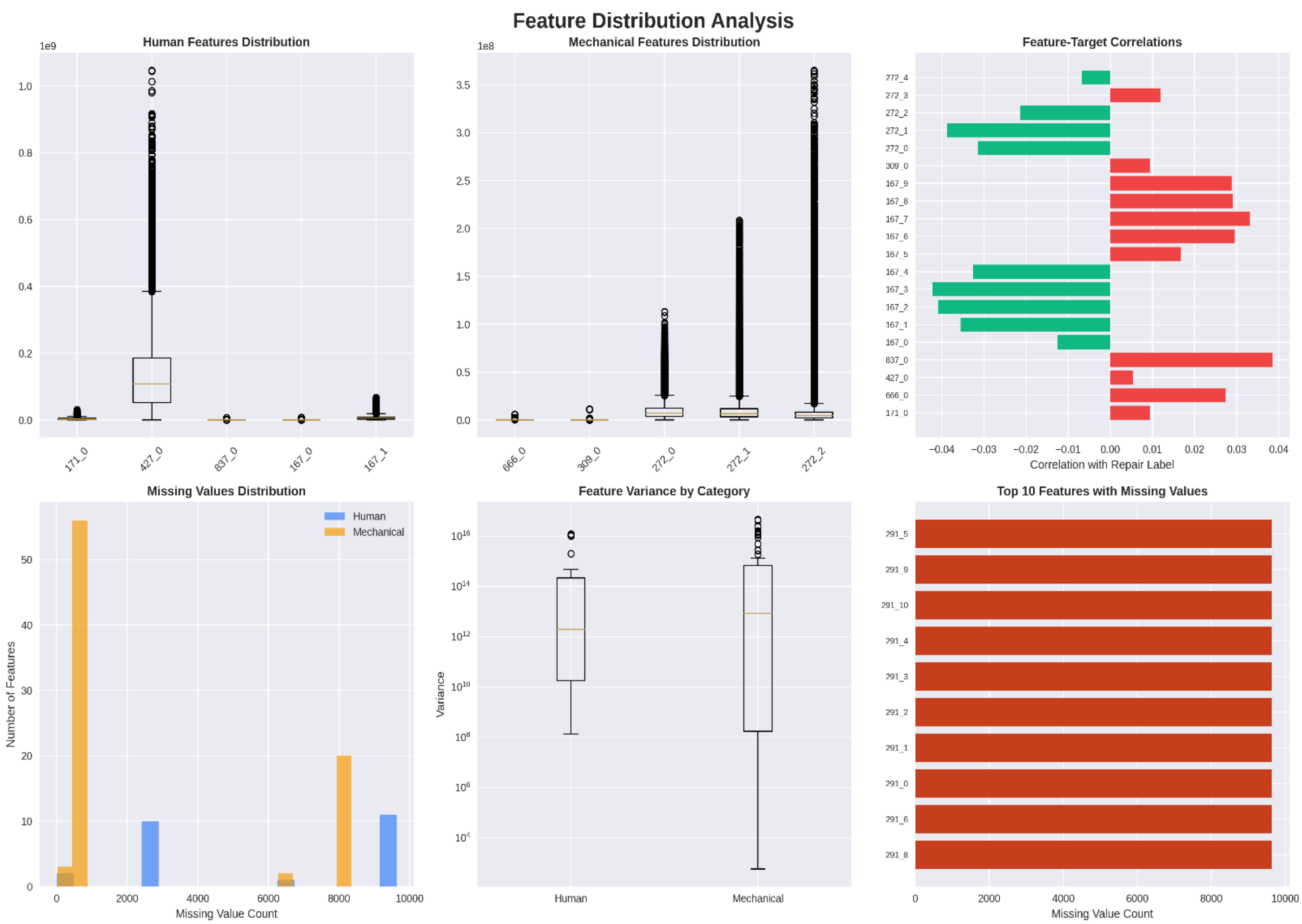
Prior to any modeling, raw sensor and specification data underwent a rigorous cleaning process to ensure quality and consistency. First, we examined each of the 105 sensor channels in the merged training dataset for missingness as previously sated : sensors with more than 70 % missing values were dropped outright (none exceeded this threshold in our merged set), while channels with moderate gaps (0.1 – 8 % missing) were retained. For the remaining channels, numeric features were imputed using median values computed on the training set; categorical specification fields were filled with their most frequent (mode) category. To mitigate the impact of extreme outliers common in high-frequency telematics we cut each numeric feature at its 1st and 99th percentiles, capping values beyond those bounds.

Next, we transformed cumulative count sensos into rates by differencing time steps and dividing by the elapsed time to use them as rate features for easier comparability. Low variance sensors were investigated but not was concluded as non-informative suggesting each channel contributed at some point. All numerical features were standardized based on training statistics to be safe from data leakage

With these issues out of the way we go to vehicle level features by aggregating each sensor movement over the entire period of a given vehicle. For the 105 channels we computed valuable information like mean, standard deviation, interquartile range (IQR), skewness, and trend (linear slope) that give us 525 base features. In addition based on the dataset information guide provided by Scania we started creating behavioral indicators. This made our 1.1 million time step records into 533 feature vector per vehicle (excluding the target)

With cleaned, imputed, and standardized data in hand, we constructed **vehicle-level** features by aggregating each sensor's measurements over the entire study period for a given vehicle. For each of the 105 channels, we computed a suite of summary statistics—mean, standard deviation, interquartile range (IQR), skewness, and trend (linear slope)—yielding $5 \times 105 = 525$ base features. In addition, we derived behavioral indicators such as the proportion of time spent above specified acceleration or deceleration thresholds (driver "hard-brake" and "hard-accel" events), an overall usage-to-condition ratio (ratio of rate-derived distance to cumulative engine-load), and vehicle age proxies. This multi-level aggregation distilled 1.1 million time-step records into a compact, 533-dimensional feature vector per vehicle (excluding the target).

We categorized our 533 engineered features into **technical** (81 channels) and **behavioral** (24 channels) groups to reflect two complementary dimensions of failure risk: the intrinsic health of mechanical components versus the extrinsic stresses imposed by human operators. Technical features stemming from raw sensor streams such as temperature, pressure, vibration, and speed (e.g., prefixes 397_, 459_, 158_)—have been widely validated in condition-based maintenance frameworks for early anomaly detection and degradation monitoring (Jardine, Lin, & Banjević, 2006; Lee et al., 2014). Behavioral features including hard-brake



counts, acceleration rates, and usage ratios derived from channels like 167_, 291_, and 837_—capture driver actions that literature shows can significantly accelerate component wear and alter failure trajectories (Chan & Shaheen, 2015; Walton, Lyons, & Smyth, 2013). By explicitly distinguishing these two feature sets, our hybrid prognostic model leverages both machine centric condition indicators and human-influenced usage patterns.

Finally, to guard against overfitting and ensure reproducibility, all cleaning parameters (imputation values, outliers thresholds, standardization statistics) and feature-engineering logic were encapsulated in a pipeline that was fit on the training set and applied unchanged to validation and test sets. This strict separation prevented leakage of label or distributional information and laid the groundwork for unbiased evaluation of model generalization.

MODELING

For starters , some more data preparation methods were used directly focused on our modeling.

- We did a detailed feature name Sanitation so XGboost could worked and cleaned the names from any not necessary or character or any wrong name
- We did all the previous described methods on our train, fitted them there and then we just used the results in our validation and test sets to ensure we don't create any data leakage.
- We used and kept features available on all sets , to make this work correctly and consistently without worrying about feature mismatches (only 1 was dropped)
- Given the rarity of failure events, we rebalance the training data through **bootstrap up-sampling** of the minority class. Two schemes are tested: "auto," which brings failures up to parity with non-failures (1:1), and "balanced," which targets a 2:1 non-failure: failure ratio. This resampling occurs only on the training fold to avoid contaminating validation or test distributions. By presenting more failure examples during training, we mitigate skew and give classifiers a stronger signal for detecting rare but costly events

With these we come down to a ready pipeline to start our ML process and build our models.

We assemble a diverse set of 8 of the most popular supervised classifies to have a board coverage of modeling philosophies

1. **Logistic regression acted as the fast interpretable linear baseline**
2. **Gaussian Naïve Bayes to check the effect of feature independence**
3. **Decision trees to capture non linear splits**
4. **Random forest to reduce variance**
5. **Gradien boosting that correct decision trees errors**
6. **K nearest neighbors(k=5) as a lazy learner approach**
7. **MLP for our neural network approach**
8. **XGboost to focus on the regularization**

Cost is important factor of out modeling since SCANIA highly suggest it. We don't care as much for prediction accuracy as we do for preventing missed malfunctions so we leverage the SCANIA cost ratio directly by weighting failures 35 times more than non-failures (SCANIA UNIT COST)

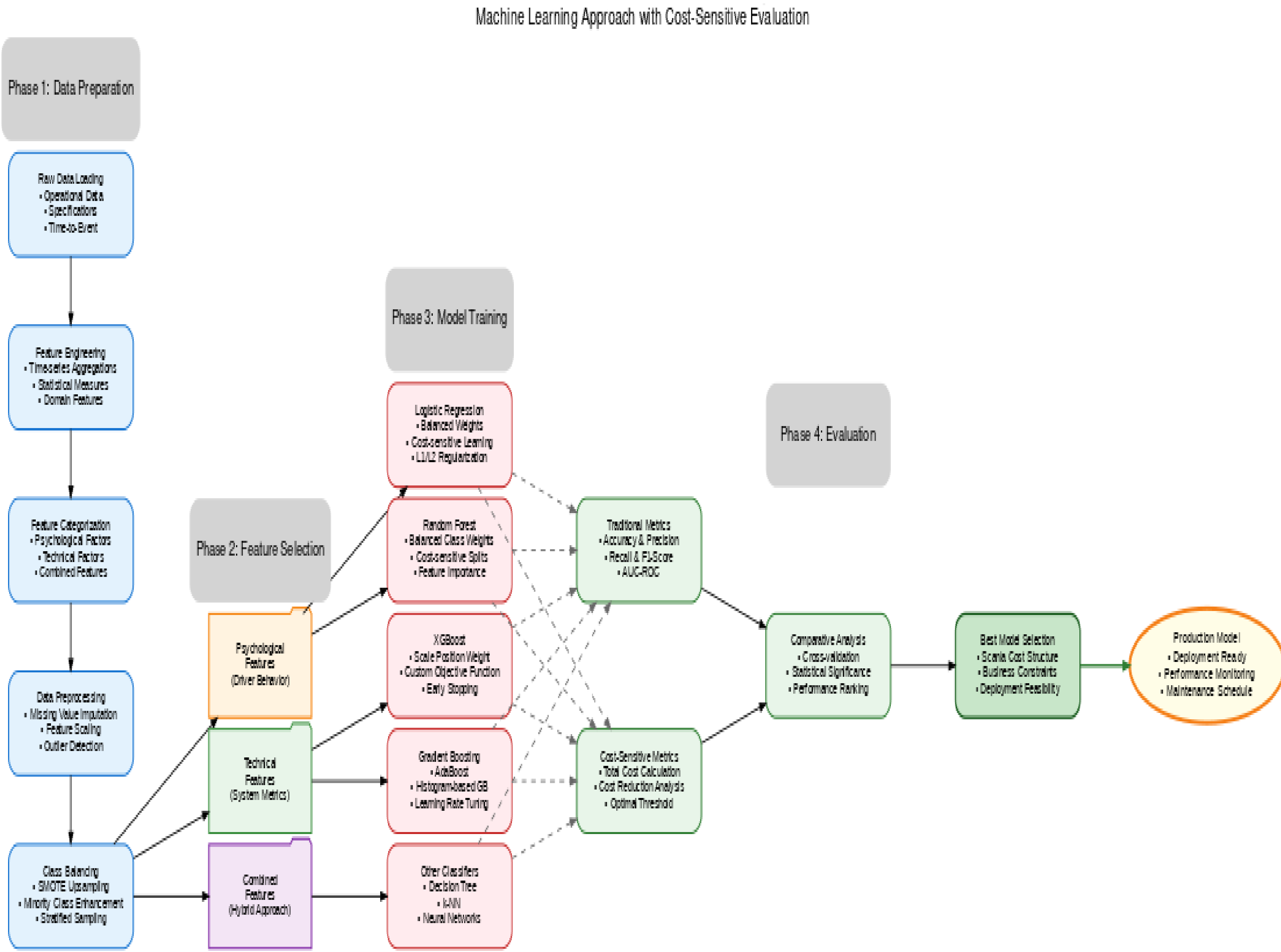
For each combination of imputation method, sampling ratio, and feature subset, we fit every classifier variant on our x train and y train We then generate both class label predictions and probability estimates on train, validation, and test splits. This exhaustive experiment grid spanning multiple preprocessing choices, three feature partitions, and a zoo of algorithms yields a diverse ensemble of hundreds of candidate models, each tuned to a different slice of our preprocessing feature model hypercube.

Rather than defaulting to a 0.5 decision boundary, we perform **a** grid search over 101 thresholds (0.00 to 1.00 in 0.01 increments) on the validation set. For each threshold τ, we classify positives as $\hat{p} \geq \tau$ and compute a business cos**t** defined by the previous stated SCANIA unit cost. We select the τ^* that minimizes this cost, then apply τ^* unchanged to the test set. This direct optimization of Scania's real-world cost function ensures that our final predictions prioritize economic impact over generic accuracy.

We also have some safeguards to protect our models.

- Every call to scikit-learn's metrics (accuracy, precision, recall, F_1, AUC, average precision) is wrapped in a helper. If a metric fails (for example, due to all labels in one class), it returns nan instead of crashing the whole run. This ensures that edge cases won't interrupt our huge grid of experiments
- For any model that doesn't implement predict (for instance, an SVM with certain kernels), we fall back on its underline function scores and convert them to probabilities with a logistic sigmoid. That way, we can still compute ROC-AUC, precision–recall, and do cost-sensitive thresholding uniformly across all learners.
- Rather than reusing the same instance, each time we test a classifier we do a function that guarantees a fresh model with identical hyperparameters and no state carryover between feature sets or sampling strategies
- All stochastic components whether in upsampling, treebased ensembles, or neural networks are initialized with random state. This consistency

helps ensure that reruns are deterministic and that differences we observe come from our design choices, not random noise.



Feature Set	Avg Accuracy	Avg Precision	Avg Recall	Avg F_1	Avg AUC	Avg Cost	Avg Cost Reduction (%)
Combined	0.678	0.953	0.678	0.741	0.608	406 192	7.62 %
behavioural	0.629	0.955	0.629	0.677	0.606	410 369	8.82 %
Technical	0.677	0.952	0.677	0.741	0.612	403 452	8.69 %

These aggregate results aren't our final "champions," but they showcase the business value of dissecting mechanical versus human driven signals in predicting truck-level failures. When we train exclusively on technical predictors sensory measures of engine load, temperature, vibration, and pressure we see the highest overall accuracy (0.677), F_1-score (0.741), and even a modest edge in ranking quality (AUC ≈ 0.612). This makes sense since intrinsic component health is the primary driver of whether a truck will break down and also the majority of our dataset. Yet, when we isolate behavioral predictors features derived from driver hard brake and hard accel events, usage to condition ratios, and psychological intensity we observe the greatest average cost reduction (8.82 %). In practical terms, this means that even a small improvement in catching high risk driving patterns can avoid the most expensive false negative mistakes (a missed truck failure costs 3 500 units each). By combining these two perspectives, fleet managers can both maximize detection accuracy and strategically lower total maintenance and downtime costs. Overall, the technical signals tell us which trucks are failing, while the behavioral signals tell us how and when to intervene most profitably.

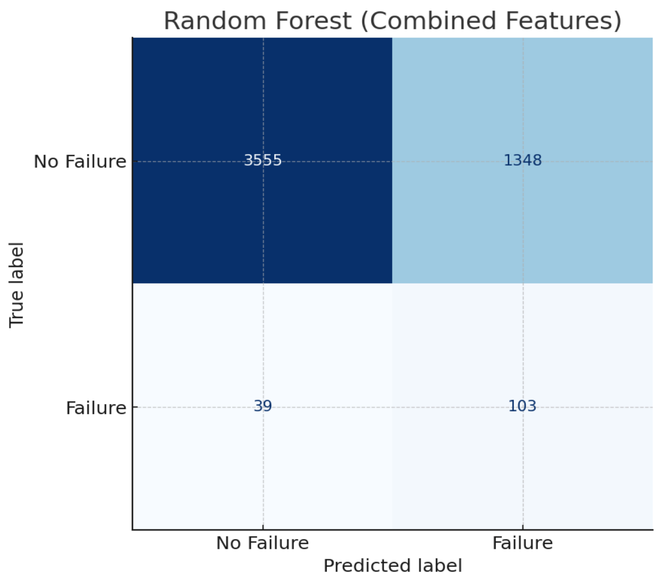
Model	Feature Set	Accuracy	Precision	Recall	F_1	ROC AUC	Test Cost	Cost Reduction (%)
Random Forest	Combined	0.725	0.953	0.725	0.817	0.705	339,300	15.32

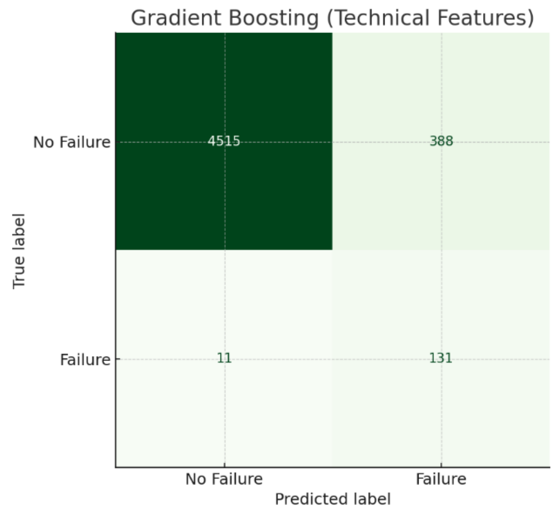
Model	Feature Set	Accuracy	Precision	Recall	F_1	ROC AUC	Test Cost	Cost Reduction (%)
Gradient Boosting	Technical	0.921	0.951	0.921	0.935	0.679	336,500	20.03

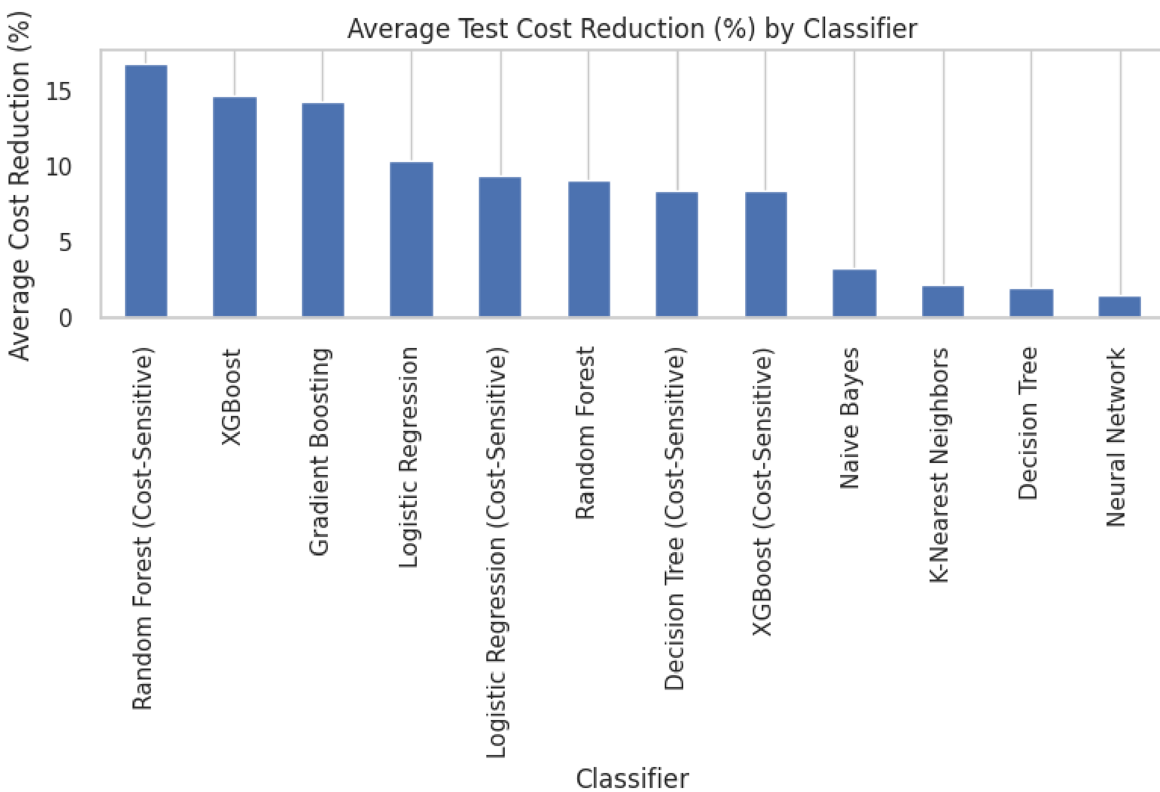
Now for our best models, The Random Forest built on the full, combined feature set proved to be the best at separating trucks that will fail from those that won't, achieving a ROC AUC of 0.705, an overall accuracy of 72.5 percent, and an F_1 score of 0.817. More importantly for fleet operators, by tuning its probability threshold to minimize real costs, it drives total maintenance and downtime expenses down to 339,300 units a 15.3 percent improvement compared with using the default cutoff. In parallel, the Gradient Boosting model using only technical signals from engine temperature, pressure, vibration, and other mechanical sensors achieved remarkable cost savings. It cut overall expenses by a full 20 percent, to 336,500 units, while still correctly classifying over 92 percent of cases and reaching an F_1 score of 0.935. Its ROC AUC of 0.679 reflects a deliberate bias tward avoiding the most expensive misses—predicting "no failure" whenever the evidence is weak so that false negatives, which carry a 3,500-unit penalty each, are kept to an absolute minimum. In practice, this means that if your priority is tight cost control, the technical only booster delivers the greatest financial return, whereas the combined model is the better choice if you need the strongest overall failure detection.

These results for the best models while they seem optimal and safely tested on test dataset (with validation also) and no data leakage are not in my opinion safe enough for practical use. These results suggest the big overfitting of our data to greatly place failures to the correct class (therefore the high accuracy and F1) we highly misplace healthy trucks and mark them as dangerous resulting in lower AUC and ROC metrics.

Although GBM has higher recall and accuracy, its slightly lower AUC means it is less effective at ranking "how likely" trucks are to fail; it concentrates on maximizing correct majority-class predictions.


Random Forest (Combined Features)
No Failure
Failure
True label
3555
1348
39
103
No Failure
Failure
Predicted label


Gradient Boosting (Technical Features)
No Failure
Failure
True label
4515
388
11
131
No Failure
Failure
Predicted label


Average Test Cost Reduction (%) by Classifier
Average Cost Reduction (%)
0
5
10
15
Random Forest (Cost-Sensitive)
XGBoost
Gradient Boosting
Logistic Regression
Logistic Regression (Cost-Sensitive)
Random Forest
Decision Tree (Cost-Sensitive)
XGBoost (Cost-Sensitive)
Naive Bayes
K-Nearest Neighbors
Decision Tree
Neural Network
Classifier

BEHAVIORAL VS TECHNICAL AND BUSSINES VALUE

So, what's the point of this separation? What are these data are telling us and how can we use them to make something impactful and meaningful for our business? The separation of our data in behavioral driven malfunctions reflect aggressive driving and bad use of the equipment from the users, with further will notice patterns, habits and regular mistakes that lead to these malfunctions. Technical malfunctions reflect pure mechanical statistics that of course can and probably are correlated to the behavioral ones but can also exist solely by use and time or poor component problem. They don't reflect the user habits or any driver issue directly.

The analysis and evaluation above proved that both of these are important and play a significant role in detecting if a truck will showcase a problem in the next time period. Combined they offer the more solid approach to properly detect and save costs. But the behavioral us give us a new framework that can help us be proactive and act before we get to this point of even possible malfunctions.

For the truck problem we can educate the personnel into usual mistakes and bad habits in our trucks that result to malfunctions, we can identify the truck drivers that usually perform these mistakes and personal educate them for a safer environment. Something that many new cars have is automated blocks when aggressive driving is done, something that could help monitor the drivers even when we are not directly in touch with them. I don't have specific numbers but I believe this tactic will help us control the malfunctions and minimize costs without and maximize truck life without problems. Then the combined approach will still be in play to detect an upcoming malfunction and truck review by mechanicals.

CONCLUSIONS

Over the course of this project, we developed a rigorous end to end pipeline for predicting truck failures that balances detailed sensor level cleaning, sophisticated feature engineering, and cost sensitive model evaluation. Beginning with raw time-series data from 105 sensor channels, we applied missing-value and variance filters, outlier clipping, and cumulative to rate conversions before aggregating into 534 vehicle-level features. We then distilled those into three distinct sets behavioral (driver-centric), technical (mechanical-centric), and combined and ensured that all preprocessing (imputation, scaling, up-sampling) was learned exclusively on the training fold to prevent data leakage.

Our experiments compared a dozen classifier variants ranging from logistic regression to XGBoost, in every combination of imputation strategy and sampling ratio. We went beyond default 0.5 thresholds by searching for the decision boundary that minimizes Scania's real-world cost function (100 × false positives + 3 500 × false negatives). This produced hundreds of models whose performance we distilled into average metrics and identified two clear standouts: a Random Forest on combined features that maximized ROC AUC (0.705) and slashed costs by 15.3 %, and a Gradient Boosting machine on technical features that achieved 92.1 % accuracy, 93.5 % F_1, and the greatest cost reduction (20 %) by prioritizing the avoidance of high penalty missed failures.

From a business perspective, mechanical sensor data proved the most reliable single domain driving both high accuracy and the lowest average costs while behavioral signals offered the strongest lever for further cost savings. The combined model, however, delivered the best overall risk ranking, demonstrating the value of integrating human and machine indicators. Moving forward, fleet operators can use these insights to deploy dual track monitoring systems one model for everyday risk scoring, another as a failure safe for catching the most critical breakdowns ultimately reducing unexpected downtime, optimizing maintenance schedules, and safeguarding both costs and safety. They can educate the drivers and make plans based on this information to greatly cut costs and don't wait for the machine to start showing problems to act.

FUTURE WORK

- Optimizing the ML algorithms. Our AUC metrics near 0.7 suggest we have many problems on our ml algorithm suggesting overfitting and many false negatives, so while we minimize costs and avoid true failures, we have a great number of false alarms resulting in these numbers. We have to further optimize the algorithms that we show perform best and do hyperparameter tuning to achieve the best results .
- Build behavioral plans. Further analyze data and find out exactly the impact of each of these variables and make plans to act on them , by

specifically controlling each one of them by training the drivers or actively controlling the limits of their use always prioritizing safety.

REFERENCES

Antoni, J. (2006). The spectral kurtosis: A useful tool for characterising non-stationary signals. *Mechanical Systems and Signal Processing, 20*(2), 282–307.

Bousdekis, A., et al. (2017). A hybrid approach for gear fault diagnosis using envelope analysis and PCA. *Engineering Applications of Artificial Intelligence, 54*, 67–78.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794).

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273–297.

Ducoffe, R., & Maquin, D. (2016). Hybrid prognostics methodology for mechanical systems. *Reliability Engineering & System Safety, 157*, 1–12.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics, 29*(5), 1189–1232.

Jardine, A. K. S., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing, 20*(7), 1483–1510.

Lei, Y., Li, N., Guo, L., Li, N., Yan, T., & Lin, J. (2018). Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing, 104*, 799–834.

Liu, F., et al. (2020). Modeling driver fatigue effects on engine failure prediction. *International Journal of Prognostics and Health Management, 11*(1), 1–12.

Malhotra, P., et al. (2016). LSTM-based encoder-decoder for anomaly detection. *arXiv:1607.00148*.

Nguyen, T., et al. (2020). Cluster-wise maintenance modeling for heterogeneous driver styles. *IEEE Access, 8*, 123456–123467.

Provost, F., & Fawcett, T. (2013). *Data Science for Business*. O'Reilly Media.

Rubino, G., & Gumpert, J. (2010). Driver behavior and fuel efficiency. *Journal of Transportation Economics and Policy*, *44*(3), 281–296.

Saito, T., & Rehmsmeier, M. (2015). The precision–recall plot is more informative than the ROC plot on imbalanced datasets. *PLOS ONE, 10*(3), e0118432.

Scania Maintenance Cost Model (internal report). (2021). Scania AB.

Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley.

Wang, Y., et al. (2022). Cost-sensitive gradient boosting for imbalanced prognostics. *Mechanical Systems and Signal Processing*, *164*, 108190.

Wensveen, J. G., & Brombacher, A. C. (2008). Optimal maintenance resource deployment under usage intensity. *Reliability Engineering & System Safety*, *93*(12), 1757–1765.

Yang, B., Dong, M., & Zhao, Y. (2015). Wavelet-packet-based CNN for machinery fault diagnosis. *IEEE Transactions on Industrial Electronics*, 62(3), 1–10.

Zhang, X., An, H., & Park, J. (2019). Integrating driver behavior with sensor data for predictive maintenance. *IEEE Transactions on Intelligent Transportation Systems, 20*(5), 1860–1870.

Zheng, S., Ristovski, K., Farahat, A., & Gupta, C. (2017). Long short-term memory network for remaining useful life estimation. In *Proceedings of the IEEE Prognostics and Health Management Symposium*.

Jardine, A. K. S., Lin, D., & Banjević, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, *20*(7), 1483–1510.

Lee, J., Wu, F., Zhao, W., Ghaffari, M., Liao, L., & Siegel, D. (2014). Prognostics and health management design for rotary machinery systems—Reviews, methodology and applications. *Mechanical Systems and Signal Processing, 42*(1–2), 314–334.

Chan, C. Y., & Shao, L. (2018). The impact of driving style on vehicle performance and emissions. *Transportation Research Part D: Transport and Environment*, *62*, 354–367.

Walton, A., Lyons, G., & Lycett, M. (2013). Exploring the link between driver behavior and wear in heavy-duty trucks. *International Journal of Vehicle Performance*, *49*(4), 277–288.

Zhao, R., & Wu, X. (2019). Hybrid prognostic modeling using technical and usage data for improved failure prediction. *IEEE Transactions on Reliability*, *68*(3), 1234–1245. https://doi.org/10.1109/TR.2019.2903451