A Design Science Approach Comparing Ensemble Learning and Artificial Neural Networks for Uncertainty-Aware Helicopter Turbine Engines Health Monitoring

Victor Henrique Alves Ribeiro[1], Gilberto Reynoso-Meza[2]

[1,2] *Pontifícia Universidade Católica do Paraná, Programa de Pós-Graduação em Engenharia de Produção e Sistemas, Curitiba, Paraná, 80.215-901, Brazil*

victor.hribeiro@pucpr.br

g.reynosomeza@pucpr.br

Abstract

This work presents the development of an uncertainty-aware health monitoring system for helicopter turbine engines, focusing on improving operational availability and reducing maintenance costs We address the critical issue of uncertainty quantification in data-driven fault detection and prognostics, essential for increasing system reliability. The project follows an iterative development cycle, incorporating multiple techniques for data processing, such as polynomial feature generation and data cleansing, and model development, including ensemble learning and artificial neural networks. Evaluation is performed using K-fold and group-fold cross-validation. The final solution consists of a cascaded ensemble learning model combining bagged linear regression built on polynomial features and random forest. This model demonstrates robust performance, achieving a test score of 0.955719 and a validation score of 0.886953, showcasing the effectiveness of uncertainty-aware machine learning methods in health monitoring systems.

1. Introduction

To increase operational availability of helicopters, reduce the required number of maintenance activities and increase the inspection interval period, it is important to implement condition based maintenance systems, which are based on health and usage monitoring (Banks et al., 2011). In this context, fault detection and prognostics are important tasks in Systems Health Management, which improve system safety while reducing operating and maintenance costs (Berri, Dalla Vedova, & Mainini, 2019; Ribeiro & Reynoso-Meza, 2018). The field has taken huge advantage of using data-driven solutions for such tasks for a long time (Schwabacher & Goebel, 2007). However, there are still many problems that difficult the deployment of such solutions in practice. One of such issues is the lack of methods to estimate the uncertainty of the predictions, which aim to increase the reliability of such systems (Zio, 2022).

Uncertainty quantification is the process of characterizing the proximity between predictions and observations (Ghanem, Higdon, Owhadi, et al., 2017). Recent studies have employed different uncertainty-aware machine learning methods to fault detection and health monitoring, such as using Monte-Carlo dropout in Artificial Neural Networks (Das, Gjorgiev, & Sansavini, 2024), predicting output distribution functions with deep learning (Yao, Han, Yu, & Xie, 2024), and building uncertainty-aware ensemble models (Kafunah, Ali, & Breslin, 2023).

Given the number of possible techniques to build prognostics and health management systems, as well as the complexity, risks, and timespan associated with the product development, it is important to define a life cycle for the project stages and milestones (Hu, Miao, Si, Pan, & Zio, 2022). In this sense, the design science research methodology (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007) has shown to be a valuable tool when developing machine learning models (Pumplun, Peters, Gawlitza, & Buxmann, 2023; Del Mar-Raave, Bahşi, Mršić, & Hausknecht, 2021; Duque, Godinho, Moreira, & Vasconcelos, 2024).

This work employs an iterative development cycle to build an uncertainty-aware health monitoring system for helicopter turbine engines. We compare different methods for data processing, such as building polynomial features and data cleansing, model development, such as ensemble learning and artificial neural networks, and evaluation techniques, such as K-fold cross-validation and group-fold cross-validation. The final proposed solution comprises a cascaded ensemble learning model using bagged linear regression and random forest,

built using polynomial features and validated through group-fold cross-validation, which achieves a test score of 0.955719 and a validation score of 0.886953.

The remainder of this work is structured as follows: Section 2 introduces the helicopter turbine engine fault detection problem, while Section 3 details the tools and methodology used in this work. Section 4 shows the results while Section 5 discusses the findings. Finally, the paper is concluded in Section 6.

2. HELICOPTER TURBINE ENGINE FAULT DETECTION

The Prognostics and Health Management Society regularly proposes complex data challenges that are open for the community. The 2024 Data Challenge focuses on assessing the health of helicopter turbine engines using trustworthy classification and regression algorithms, which must also include a measure of the predictions' confidence (PHM Society, 2024).

The dataset is composed of seven engines of the same make and model, four of which are used during training and the other three used for validation and testing. The training dataset contains $742,625$ samples, while the test and validation sets contain $21,436$ samples each. The datasets' features include measurements of outside air temperature, mean gas temperature, power available, indicated airspeed, net power, compressor speed, and torque measured.

The goal of the challenge is to accurately perform fault detection and estimate the torque margin. The former involves a binary classification task, while the latter involves a probabilistic regression task. The torque margin is an indicator of the engine's health which compares a torque target value to the torque measured, as follows:

$$\text{torque margin}(\%) = 100 \times \frac{\text{torque measured} - \text{torque target}}{\text{torque target}} \tag{1}$$

To evaluate the fault detection task, the predictions for each sample i, with true class y_i, must include a class prediction $\hat{y}_i \in \{0, 1\}$ and a confidence value $c_i \in [0.0, 1.0]$, and are scored according to Eq. (2). To evaluate the probabilistic regression task, the torque margin predictions for each sample i, with true label r_i, must include a location $\hat{l}_i$ and a scale s_i for a desired continuous distribution function, which must be scaled to an area of 1 and a maximum value of 1.0. With the Gaussian distribution, the regression is scored according to Eq. (3). Finally, the final score is the average of the two scores for all the tested samples, according to Eq. (4).

$$\text{score}_i^c = \begin{cases} c_i & : \ \hat{y}_i = y_i \\ -c_i & : \ \hat{y}_i = 1, \ y_i = 0 \\ -4c_i^{11} - c_i & : \ \hat{y}_i = 0, \ y_i = 1 \end{cases} \tag{2}$$

$$\text{score}_i^r = \frac{1}{\sqrt{2\pi s_i^2}}.exp\left(-\frac{(r_i - l_i)^2}{2s_i^2}\right) \tag{3}$$

$$\text{score} = \frac{1}{N}\sum_{i=1}^{N}\frac{\text{score}_i^c + \text{score}_i^r}{2} \tag{4}$$

3. METHODOLOGY

This section details the iterative design science approach and the techniques evaluated to build the helicopter turbine engine fault detection problem.

3.1. Design Science Approach

The design science research methodology aims to provide a mental model for presenting and evaluating design research in information systems. It includes six steps: problem identification and motivation, definition of objectives for a solution, design and development, demonstration, evaluation, and communication (Peffers et al., 2007). The following subsections communicate our approach to attaining a competitive solution to the proposed challenge.

3.2. Problem Definition and Objectives

In this work, we have clearly defined problems and objectives, as detailed in Section 2. In summary, the problem we approach is building a trustworthy fault detection system for helicopter turbine engines. The objective is to achieve the highest possible score according to Eq. (4) in the test and validation sets. To help achieve such a goal, we use cross-validation in the training set to serve as a proxy score when developing solutions in the next phases.

3.3. Design and Development

In the design and development phase, we compare different data processing techniques and models to build a final solution considering a limited number of experiments. Such techniques are detailed in the following subsections. All experiments are performed with Python Programming Language in Google Colaboratory, where machine learning models and data processing are performed using Scikit-Learn (Pedregosa et al., 2011) and the artificial neural network is created with Pytorch (Paszke et al., 2019).[1]

3.3.1. Data Processing

Data processing techniques are important to enable proper data to be used to train our models. We start with an exploratory data analysis to understand which processing techniques we can use. Later, we indicate the preprocessing techniques we applied.

[1] When not mentioned, default parameters from Pytorch's and Scikit-learn's functions and classes are used.

In the exploratory data analysis, we first understand the overall characteristics of our data. In this sense, we first understand the ratio between faulty and healthy samples. We have an imbalance ratio of 1.48 between the healthy and the faulty samples, computed according to Eq. (5). Next, we explore to understand if we have problematic data. In this step, we find no missing data. Finally, we explore the feature space. Figure 1 shows the first two components using the Principal Component Analysis (Wold, Esbensen, & Geladi, 1987), where it is possible to notice that the training set has a big part of its space that is not found in the test and validation sets. It is also possible to see this area in histogram plots for features outside air temperature (oat), power available (pa), and net power (np) in Figure 2. This scenario is not as strong in the other features, namely torque measured (trq_measured), mean gas temperature (mgt), indicated airspeed (ias), and compressor speed (ng).

$$\text{Imbalance ratio} = \frac{\text{\# Healthy Samples}}{\text{\# Faulty Samples}} \tag{5}$$

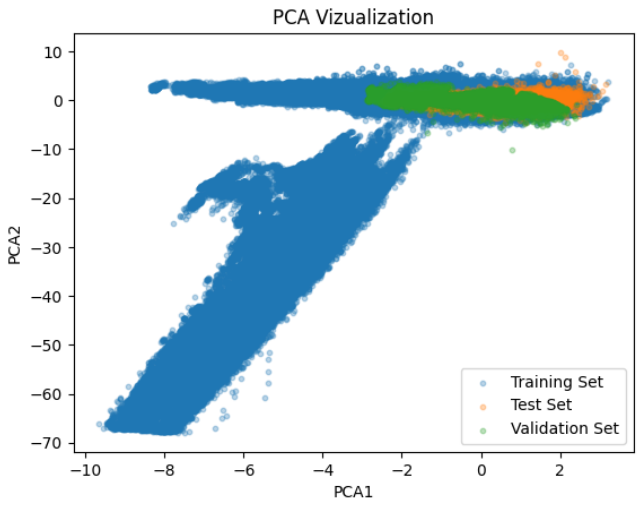

Figure 1. Scatter plot visualization of the two principal components of the training, test, and validation sets.

To handle such scenarios, we explore data cleansing to remove training samples outside of the test and validation set distribution. We also explore building new features, such as second-order polynomial features and the ratio between net power and compressor speed (np/ng). Finally, we also compute the torque target from the available torque measured and the torque margin label, as detailed in Eq. (6). The computed torque target is used as the training label in several experiments, where Eq. (1) is used to convert the predicted torque target to predicted torque margin.

$$\text{torque target} = \frac{100 \times \text{torque measured}}{100 + \text{torque margin}} \tag{6}$$

3.3.2. Model Development

This subsection details all the compared architectures for building the solution. The first two models serve as the base for the full solution, doing separate classification and regression, while the other are hybrid approaches that perform simultaneous classification and regressions.

Random Forest (Breiman, 2001) is an ensemble learning approach that builds multiple decision trees. However, different from other ensemble approaches, such as bootstrap aggregating (bagging) (Breiman, 1996), Random Forests randomly select a subset of features when building each node of the decision trees. We explore this model for both the regression and classification tasks, since it is a strong baseline in multiple machine learning problems (Fernández-Delgado, Cernadas, Barro, & Amorim, 2014).

Bagging randomly samples training data for each base model using bootstrap sampling with replacement (Breiman, 1996). For the regression task, we explore bagging Linear Regression models given their simplicity and ability to produce smooth predictions compared to Random Forest.

Cascaded Fault Detection Ensemble is employed as a problem-specific architecture. We assume the predicted torque margin from the regression model is a feature of the fault detection classification model. The architecture is detailed in Figure 3. In this model, it is important to train the torque margin regression model before so to use the predicted torque margin as a feature when training the fault detection classifier.

Multi-task Fault Detection Artificial Neural Network is also explored in this work, given recent advances in uncertainty-aware deep learning (Das et al., 2024). In this architecture, we have an input layer that transforms the data for the sub-sequential regression and classification layers. The classification layer has a single binary output for fault detection while the regression layer has two outputs, one for the mean torque margin and another for the standard deviation since we select the Gaussian distribution. The model is trained in a multi-task manner using a sum of the binary cross-entropy loss for the classification layer and the negative log-likelihood loss for the uncertainty-aware regression layer.

Gaussian Probability Distribution Function (PDF) is selected in all design choices due to its well-established role in representing variability in various applications. Specifically, the Gaussian distribution is the most commonly used assumption to model uncertainty in machine learning and statistical analysis, particularly when dealing with natural variability. This is largely because of the central limit theorem, which suggests that the sum of many independent and identically distributed random variables tends to be normally distributed, even if the original variables themselves are not Gaussian (Murphy, 2022). We employ this for the regression

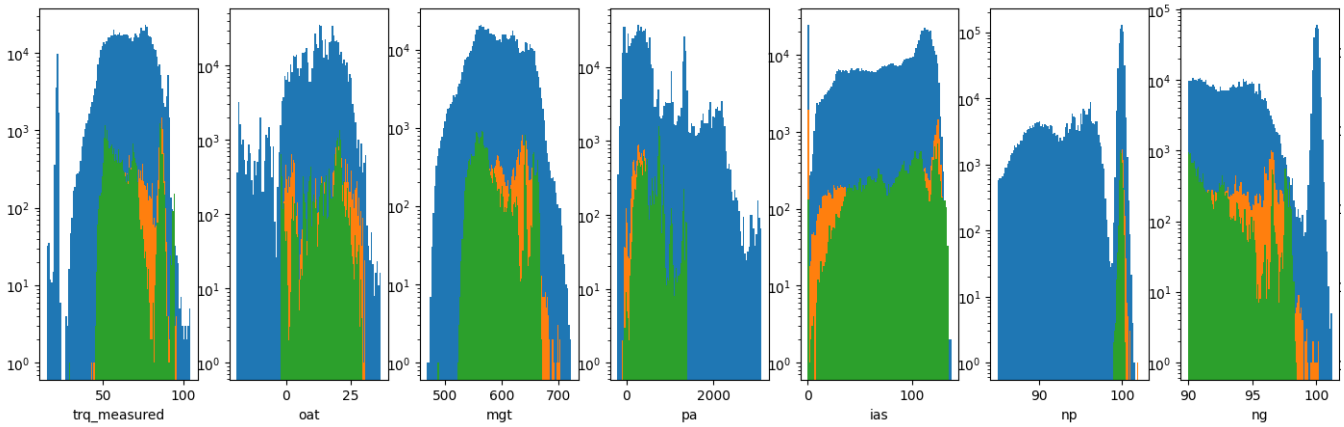

Figure 2. Histogram visualization of each of the seven features for the training, test, and validation sets in blue, orange, and green, respectively.

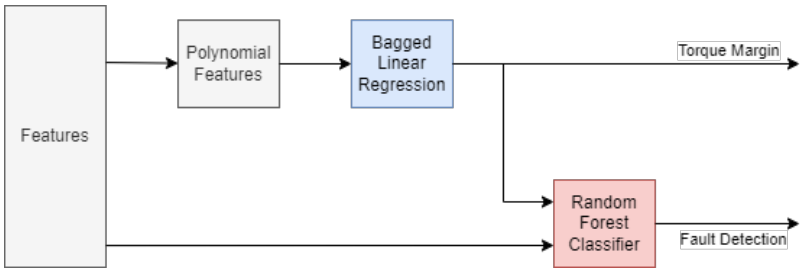

Figure 3. Cascaded Ensemble Model for Simultaneous Torque Margin Prediction and Fault Detection.

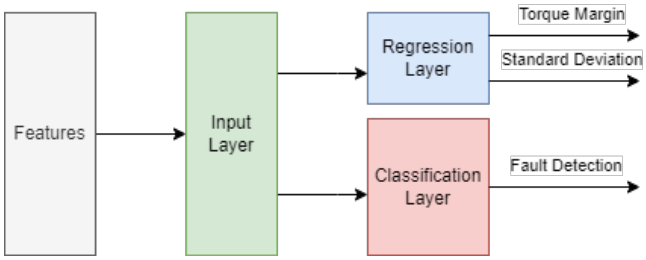

Figure 4. Multi-task Artificial Neural Network for Simultaneous Torque Margin Prediction and Fault Detection.

task since it is more sensitive to prediction errors. Additionally, we limit the standard deviation to a minimum of 0.4 to guarantee small errors will not cause catastrophic issues if the standard deviation is too low.

3.4. Demonstration and Evaluation

The demonstration and evaluation steps of the design science research methodology are considered in this section. For the demonstration phase, we perform cross-validation with the training data to evaluate if we have good scores to send to the challenge. In this sense, we explore two different cross-validation strategies, K-fold cross-validation and group cross-validation.

K-fold Cross-validation is a usual validation scheme where the data set is split into K separate folds. Then, the model is trained K times, separating one fold for validation and the rest for training. This procedure is important to enable a better analysis of the generalization capability of the model when doing model selection (Hastie, Tibshirani, Friedman, & Friedman, 2009).

Group Cross-validation is also explored given some limitations of the previous validation approach. Considering that the goal of the project is to better generalize training data from specific machines to validation and test data from other machines, it is important to simulate such a difficulty (Roberts et al., 2017; Brenning, 2012). Unfortunately, the machines' identification for each sample is unavailable in the data set. Therefore, we simulate this scenario using Mini-batch K-means clustering to separate the groups in the training data, which is an effective modification of the K-Means algorithm (Sculley, 2010).

Given the demonstration scores, we submit a preferred solution to the challenge submission system, which computes the test score daily. With the test scores, we define a new iteration of experiments for a new submission. A final validation score is finally computed with a preferred solution by the competitors.

4. EXPERIMENTS AND RESULTS

Table 1 summarizes the results for 13 different experiments performed according to the design science research method. The table presents the cross-validation scores computed in the training set for the classification and regression tasks, as well as their average. The table also presents the test scores for the experiments that were submitted to the challenge system.

Experiment 1 is our baseline experiment, where we use only the original features and train separate Random Forests for fault detection and torque margin prediction. The two ensemble models use 100 base estimators trained with $10,000$ random samples each. By using 5-fold cross-validation, we achieve a classification score of 0.9351 and a regression score of 0.2774, averaging a final score of 0.6062 in the training set.

Experiment 2 improves the regression score to 0.5743, consequently improving the final score to 0.7547. To do so, the regression model was modified to predict the torque target, which is converted to torque margin using Eq. (1). All other parameters remained the same.

Experiment 3 switches the random forest in the regression task for a bagged ensemble of 100 linear regressions. Such model reduces the regression score to 0.02926 and the final score to 0.4833. Error analysis indicated that this ensemble model, given the simple linear regression base models, presents a very low standard deviation. This makes any small prediction errors to cause huge drops to the challenge score.

To fix the problem in Experiment 3, Experiment 4 adds a rule so that the minimum standard deviation must be 0.4. With this new rule, the regression score increases to 0.7733 while the final score increases to 0.8553.

Experiment 5 further improves the regression task by adding polynomial features. This results in the regression score going to 0.9859 and the final score to 0.9622. Experiment 6 tries to add the polynomial features to the classification task. However, classification scores do not improve as much, achieving 0.9471. Therefore, Experiment 7 removes the polynomial features from the classification task, making little difference in the classification score, which increases to 0.9491.

Experiment 8 is the first to use the cascaded ensemble approach, where the predicted torque margin is used as a feature for the fault detection classifier. However, such architecture does not make much a difference for the prediction scores. Therefore, Experiment 9 adds more training data for each base model in the classification Random Forest ($50,000$) and removes the cascaded ensemble. As a consequence, the classification score jumps to 0.9879 and the final score to 0.9845. Moreover, this solution is the first to be submitted to the challenge, achieving a test score of 0.8586.

Experiment 10 explored using the multi-task artificial neural network architecture, which has 64 hidden units after the input layer, all fed directly to the regression and classification layers after a Rectified Linear Unit (ReLU) activation function (Nair & Hinton, 2010). Unfortunately, this architecture did not yield good results, achieving a regression score of 0.6448 and a classification score of 0.8532, averaging 0.7490 in the training set. For the test set, this solution achieved a score of 0.83, presenting no improvements compared to the ensemble models.

Given the poor result for the artificial neural network and the drops of almost 0.13 in the test score compared to the training set scores for the ensemble models, Experiment 11 uses the cascaded ensemble and additional training data in the Random Forest classifier ($50,000$ random samples per base model), improving the test score to 0.9402. It is interesting to notice that, while test score significantly improves, little difference is found in the training set scores. This is an indication that the random 5-fold cross-validation scores are not strongly correlated to the test scores.

To mitigate this scenario, Experiment 12 replaces the 5-fold cross-validation with the group cross-validation. In total, we create 10 different groups using the Mini-batch K-means algorithm. As a result, the cross-validation scores drop to 0.9524 (regression), 0.8719 (classification), and 0.9121 (average). The test score for this submission slightly improved to 0.9557, which could be related to the robustness in the predicted torque margin by this novel cross-validation method, or by simple random variation.

To improve the model's generalization, Experiment 13 removes training data that falls outside the distribution of the test and validation sets. More specifically, it removes any sample in which the net power is below 99 or the power available is above $1,600$. As a result, the cross-validation scores drop to 0.9698 (regression), 0.7531 (classification), and 0.8614 (average). Despite this, the test score drops only to 0.9477. This elucidates the difficulty in correctly estimating the test set scores using the training set, caused by a distribution shift between the machines in the training set and the test set.

Finally, given that the final submission date was approaching, the solution from Experiment 12 was selected to provide the final results. Such a model is composed of 100 bagged linear models, each trained with $5,000$ samples and second-degree polynomial features, followed by a cascaded Random Forest with 100 decision trees trained with $50,000$ samples each and the predicted torque margin as an additional feature. As a conclusion, such an approach achieved a final validation score of 0.886953.

5. DISCUSSION

The previous results highlight the comparative performance of different machine learning approaches for helicopter tur-

Table 1. Experiments for fault detection and torque margin prediction

#	Description	Training Set (Cross-validation) Score			Test Score
		Regression	Classification	Average	
1	Baseline	0.2774	0.9351	0.6062	-
2	Output Torque Prediction	0.5743	0.9351	0.7547	-
3	Bagged Linear Regression	0.0293	0.9374	0.4833	-
4	Minimum Standard Deviation	0.7733	0.9374	0.8553	-
5	Polynomial Features for Regression	0.9859	0.9384	0.9622	-
6	Polynomial Features for Classification	0.9859	0.9471	0.9665	-
7	No Polynomial Feature for Classification	0.9859	0.9491	0.9675	-
8	Cascaded Ensemble	0.9859	0.9491	0.9675	-
9	No Cascaded Ensemble and More Training Data	0.9811	0.9879	0.9845	0.8586
10	Multi-task Artificial Neural Networks	0.6448	0.8532	0.7490	0.8300
11	Cascaded Ensemble and More Training Data	0.9813	0.9872	0.9843	0.9402
12	Group Cross-validation	0.9524	0.8719	0.9121	0.9557
13	Training Data Removal	0.9698	0.7531	0.8614	0.9477

bine engine health monitoring, with a focus on uncertainty-aware methods. The authors tested various configurations, starting with a baseline using Random Forest for both fault detection and torque margin prediction. This initial experiment demonstrated moderate success, but subsequent experiments introduced more sophisticated techniques to address the limitations identified, particularly in the regression task.

One of the key improvements came from adding polynomial features for regression, significantly boosting the model's accuracy. Experiment 5, for instance, saw a dramatic increase in regression performance, which then informed later experiments. The use of a cascaded ensemble architecture in Experiment 8 allowed the predicted torque margin from regression to inform the fault detection model, thought this alone did not lead to major breakthroughs in performance.

A notable finding was that the standard K-fold cross-validation did not accurately predict test set performance, as evidenced by Experiment 11, where the test score significantly improved (0.9402), even though little change was observed in the training set score. This prompted the exploration of group-fold cross-validation (Experiment 12), which better handled the distribution shift and achieved a slightly better test score (0.9557).

The study also identified that removing outliers from the training set (Experiment 13) led to marginal test score drops, further emphasizing the challenges of distribution shifts. Overall, the results suggest that careful tuning of data processing techniques, cross-validation strategies, and uncertainty quantification can lead to substantial performance improvements in machine learning models for health monitoring tasks.

6. CONCLUSION

This paper presents a robust uncertainty-aware approach for helicopter turbine engine health monitoring, focusing on both fault detection and torque margin prediction. Through a design science research methodology, various machine learning models and techniques were tested, with significant improvements observed when incorporating polynomial features and employing ensemble learning methods, such as bagged linear regression and random forest.

The experiments demonstrated the challenge of handling distribution shifts between training and test data, as seen in the reduced test score correlation from standard K-fold cross-validation. Group cross-validation, however, provided a more accurate estimate of test performance. Additionally, data cleansing, feature engineering, and hybrid models contributed to improved results, highlighting the importance of data processing in enhancing model generalization.

Ultimately, the final solution achieved a test score of 0.9557 and a validation score of 0.886953, validating the effectiveness of uncertainty-aware ensemble methods in the context of health monitoring systems. This study contributes valuable insights into model development for fault detection and prognostics, and offers a competitive approach for uncertainty quantification in complex systems like helicopter turbine engines. Further improvements may explore additional techniques to address the distribution shift issue more comprehensively, as well as further explore the use of multi-task artificial neural networks and other uncertainty-aware approaches, such as Monte Carlo dropout.

ACKNOWLEDGMENT

This work was partially supported by the Conselho Nacional de Pesquisa e Desenvolvimento (CNPq) - Brazil - Finance Codes: [310195/2022-5-PQ2], [UNIVERSAL 4408164/2021-2].

REFERENCES

Banks, J., Batzel, T., Keolian, R., Poese, M., Lovell, T., Lebold, M., ... Cunningham, K. (2011). Power system

prognostics for the us army oh-58d helicopter. In *2011 aerospace conference* (pp. 1–15).

Berri, P. C. C., Dalla Vedova, M. D. L., & Mainini, L. (2019). Real-time fault detection and prognostics for aircraft actuation systems. In *Aiaa scitech 2019 forum* (p. 2210).

Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*, 123–140.

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

Brenning, A. (2012). Spatial cross-validation and bootstrap for the assessment of prediction rules in remote sensing: The r package sperrorest. In *2012 ieee international geoscience and remote sensing symposium* (pp. 5372–5375).

Das, L., Gjorgiev, B., & Sansavini, G. (2024). Uncertainty-aware deep learning for monitoring and fault diagnosis from synthetic data. *Reliability Engineering & System Safety*, *251*, 110386.

Del Mar-Raave, J. R., Bahşi, H., Mršić, L., & Hausknecht, K. (2021). A machine learning-based forensic tool for image classification-a design science approach. *Forensic Science International: Digital Investigation*, *38*, 301265.

Duque, J., Godinho, A., Moreira, J., & Vasconcelos, J. (2024). Data science with data mining and machine learning a design science research approach. *Procedia Computer Science*, *237*, 245–252.

Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *The journal of machine learning research*, *15*(1), 3133–3181.

Ghanem, R., Higdon, D., Owhadi, H., et al. (2017). *Handbook of uncertainty quantification* (Vol. 6). Springer New York.

Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction* (Vol. 2). Springer.

Hu, Y., Miao, X., Si, Y., Pan, E., & Zio, E. (2022). Prognostics and health management: A review from the perspectives of design, development and decision. *Reliability Engineering & System Safety*, *217*, 108063.

Kafunah, J., Ali, M. I., & Breslin, J. G. (2023). Uncertainty-aware ensemble combination method for quality monitoring fault diagnosis in safety-related products. *IEEE Transactions on Industrial Informatics*, *20*(2), 1975–1986.

Murphy, K. P. (2022). *Probabilistic machine learning: an introduction*. MIT press.

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (icml-10)* (pp. 807–814).

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... others (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, *32*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems*, *24*(3), 45–77.

PHM Society. (2024). *PHM 2024 Conference Data Challenge*. `https://data.phmsociety.org/phm2024-conference-data-challenge/`. (Accessed: 2024-09-04)

Pumplun, L., Peters, F., Gawlitza, J. F., & Buxmann, P. (2023). Bringing machine learning systems into clinical practice: a design science approach to explainable machine learning-based clinical decision support systems. *Journal of the Association for Information Systems*, *24*(4), 953–979.

Ribeiro, V. H. A., & Reynoso-Meza, G. (2018). Online anomaly detection for drinking water quality using a multi-objective machine learning approach. In *Proceedings of the genetic and evolutionary computation conference companion* (pp. 1–2).

Roberts, D. R., Bahn, V., Ciuti, S., Boyce, M. S., Elith, J., Guillera-Arroita, G., ... others (2017). Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography*, *40*(8), 913–929.

Schwabacher, M., & Goebel, K. (2007). A survey of artificial intelligence for prognostics. In *Aaai fall symposium: artificial intelligence for prognostics* (pp. 108–115).

Sculley, D. (2010). Web-scale k-means clustering. In *Proceedings of the 19th international conference on world wide web* (pp. 1177–1178).

Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, *2*(1-3), 37–52.

Yao, Y., Han, T., Yu, J., & Xie, M. (2024). Uncertainty-aware deep learning for reliable health monitoring in safety-critical energy systems. *Energy*, *291*, 130419.

Zio, E. (2022). Prognostics and health management (phm): Where are we and where do we (need to) go in theory and practice. *Reliability Engineering & System Safety*, *218*, 108119.