Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 59-26 (2025) 223–228

Using Kolmogorov-Arnold Networks for an Interpretable and Continual Fault Detection of Helicopter Turbine Engines

Valerio Morelli * **Federica Paganica** * **Federico Staffolani** * **Enrico Maria Sardellini** * **Lucia Migliorelli** * **Alessandro Freddi** *

* *Department of Information Engineering, Università Politecnica delle Marche*

Abstract:
In recent years, data-driven methods have become common in fault-detection systems, with deep learning architectures – particularly Multi-Layer Perceptrons (MLPs) – achieving state-of-the-art performance. However, MLPs suffer from two critical limitations in the context of safety-critical systems: lack of interpretability and vulnerability to catastrophic forgetting under continual learning scenarios. In this paper, we present a novel approach to predictive maintenance using Kolmogorov-Arnold Networks (KANs), which are based on a different mathematical foundation than MLPs. To evaluate the efficacy of KANs for engine-health monitoring, we benchmark their performance against conventional MLPs on the PHM North America 2024 Conference Data Challenge. Our results show that KANs significantly increase model transparency – allowing the predictions to be trusted – and exhibit superior resilience to forgetting, while still achieving high test scores.
Our codes can be found at `https://github.com/MrPio/PHM_North_America_2024_Challenge`

Copyright © 2025 The Authors. This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0/)

Keywords: Fault detection and diagnosis; Intelligent maintenance systems; Prognostics and health management; Interpretable AI; Kolmogorov-Arnold networks.

1. INTRODUCTION

Fault detection systems are a cornerstone of modern predictive maintenance pipelines, especially in safety-critical applications such as helicopter turbine engine monitoring. In these settings, an incorrect prediction can trigger expensive actions – including unnecessary system shutdowns or reconfigurations – which, in turn, may compromise operational continuity and safety. Therefore, beyond accuracy, the interpretability of model outputs is a key requirement (Gohil et al., 2024; Han et al., 2024). In addition, online predictive maintenance pipelines, where environmental conditions tend to evolve over time, require resilience to the phenomenon of catastrophic forgetting, where a model struggles to remember previously acquired knowledge.

This dual need – for robustness and transparency – explains why most current solutions rely on statistical models or simple machine learning approaches (Wu et al., 2024; Alves Ribeiro and Reynoso-Meza, 2024). On the one hand, the scarcity of labeled faulty data constrains model complexity; on the other hand, interpretable models are essential to ensure trust in automated decisions. Despite the impressive predictive power of deep neural networks – particularly Multi-Layer Perceptrons (MLPs), backed by the Universal Approximation Theorem (Hornik et al., 1989) – their black-box nature makes them difficult to deploy in operational settings where *justification of predictions is non-negotiable.*

To address this, we explore an emerging architecture known as Kolmogorov–Arnold Networks (KANs) (Liu et al., 2024). KANs replace the classic weighted sums of MLPs with *trainable spline functions* placed on the edges, offering a more expressive yet interpretable alternative. By design, KANs allow inspection of individual mappings from input features to output variables, making them attractive for transparent fault detection.

To assess their validity in the fault detection context, we propose a novel KAN-based approach to tackle the PHM North America 2024 Conference Data Challenge [1], which simulates a real-world predictive maintenance scenario on helicopter turbine engines. The dataset includes sensor measurements from seven engines operating under various environmental conditions. The challenge is divided into two tasks: (i) Regression, which involves predicting the torque margin, a measure of engine performance; and (ii) Classification, which involves predicting whether the engine is in a nominal or faulty state. In addition to raw predictions, it is required to estimate the confidence or uncertainty of their outputs – a demand that further emphasizes the need for interpretable and probabilistic models.

Current state-of-the-art solutions to the PHM North America 2024 data challenge remain rooted in classical statistics and off-the-shelf machine-learning algorithms.

[1] `https://data.phmsociety.org/phm2024-conference-data-challenge/`

2405-8963 Copyright © 2025 The Authors. This is an open access article under the CC BY-NC-ND license.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2025.12.038

For regression, (Han et al., 2024) augment a polynomial regressor with empirical-error sampling to introduce probabilistic behavior, while (Alves Ribeiro and Reynoso-Meza, 2024) rely on a bagged linear model and (Ozeki et al., 2024) on a second-order polynomial regressor. (Wu et al., 2024) train a Gaussian Process regressor on a space-filling–reduced subset of the training data, and (Romano et al., 2024) achieve a great accuracy with a MLP (whose dense layers are sized [7, 256, 256, 2]).

For classification, (Han et al., 2024) employ two logistic regression classifiers; (Alves Ribeiro and Reynoso-Meza, 2024) use a random forest; and (Ozeki et al., 2024) employ a hybrid LightGBM–k-NN scheme that reconstructs temporal relationships via Euclidean distances in feature space. (Hun Park et al., 2024) build an ensemble of a deep network and XGBoost, both fed by features from a shared multi-head attention module. (Pankaj et al., 2024) exploit MATLAB's Regression Learner and `fitcauto` function to assemble AdaBoost and bagged-tree classifiers. (Wu et al., 2024) further combine CNN, MLP, XGBoost, and AdaBoost outputs via a logistic-regression meta-learner to yield a final fault-probability estimate. Finally, (Romano et al., 2024) validate a compact MLP ([7, 32, 2]) for classification, proving that minimal depth can achieve superior test performance when compared to a deeper network.

All of these methods, even if they outperform the challenge with very high test scores, do not provide a convincing transparency in the predictions. The goal of this paper is to demonstrate that KANs can tackle complex fault detection problems, such as the one addressed by the above mentioned challenge, while also delivering *resilience to continual learning*, and most importantly, *interpretable outcomes* that can support human decision-making.

2. MATERIALS AND METHODS

This section describes the details of the dataset and the preliminary operations performed on it.

2.1 Dataset

Three datasets are provided: one labeled training set and two unlabeled validation and test sets. Together, they store sensor measurements from seven identical helicopter engines – four engines in the training set and the remaining three split equally between validation and test sets. All samples have been anonymized and stripped of any time-stamp information. The training set consists of 742,625 records, while each of the validation and test sets contains 21,436 records.

Each record includes measurements of outside air temperature (oat), mean gas temperature (mgt), pressure altitude (pa), indicated airspeed (ias), net power (np), compressor speed (ng), and measured torque (trq_{msr}). The objective is to (1) predict the torque margin via regression and (2) classify the engine's health state ($0 = nominal$, $1 = faulty$). Performance is evaluated by two scoring functions – one for regression and one for classification – and the overall challenge score is the average of these two metrics. Confidence levels play a crucial role in the scoring system: in regression, overly confident predictions yield no additional reward – the score is normalized if the predicted

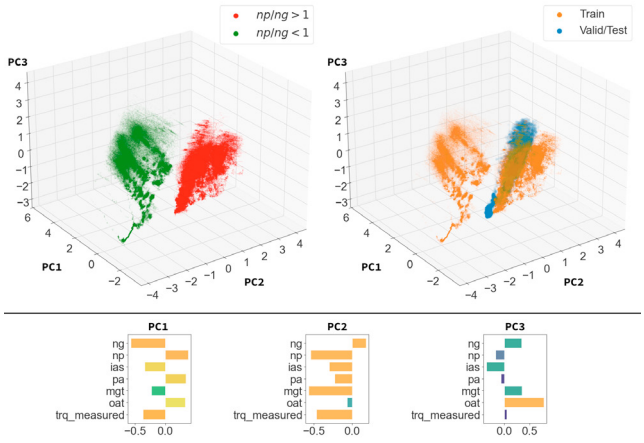

Fig. 1. Two scatter plots of the dataset in the space defined by the first 3 principal components. In the left plot, the samples are clearly partitioned into two distinct clusters by the np/ng ratio, while the right plot shows that the test and validation sets are confined entirely to one of these clusters.

PDF attains a peak value above 1 – and in classification, highly confident false negatives are strongly penalized.

2.2 Explorative Data Analysis

Prior to training, we applied an initial preprocessing step to the dataset. Following the approach of (Han et al., 2024), we performed a Principal Component Analysis (PCA) and, by following the samples distribution of the first 3 components, we decided to remove all samples for which $np < ng$, since they fall outside the domain covered by the test and validation sets (see Fig. 1). This exclusion affected approximately 0.345% of the training data.

We then evaluated multicollinearity among the measurements – a situation in which strong linear dependencies make the design matrix ill-conditioned or singular, leading to unstable parameter estimates and degraded model performance. To assess multicollinearity among our predictors, we computed the Variance Inflation Factor (VIF) for each measurement X_i, defined as

$$\mathrm{VIF}(X_i) = \frac{1}{1 - R^2},$$

where R^2 is the coefficient of determination obtained by regressing X_i on all other measurements. As a rule of thumb, a VIF below 5 indicates that a measurement does not exhibit problematic collinearity (Akinwande et al., 2015). In our analysis, however, both np and ng yielded VIF values of 40, suggesting a strong linear dependence (see Fig. 2). More specifically, we found

$$np \approx -1.06\, ng + 165,$$

with $R^2 = 0.975$.

2.3 Feature extraction

Adding derived features can enhance the accuracy of the model, especially in our case where we want to train an interpretable – and thus shallow – network. As such, we augmented the dataset with 131 engineered features. These include: (i) all polynomial combinations of the

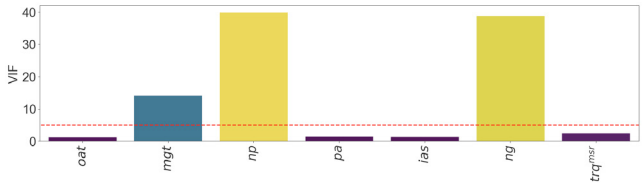

Fig. 2. VIF scores for the seven input measurements. The red dashed line marks the conventional VIF threshold; both *np* and *ng* greatly exceed this limit, indicating a severe multicollinearity problem.

base features up to third order, like (Alves Ribeiro and Reynoso-Meza, 2024); (ii) the *np*/*ng* ratio, like (Han et al., 2024); (iii) density altitude, like (Ozeki et al., 2024); and (iv) ambient air density and density-normalized versions of each base feature, like (Hun Park et al., 2024). Normalization by air density decouples the failure state of the engine from the atmospheric conditions. In fact, two identical engines may perform differently in denser air than in thinner air.

2.4 Feature selection

To mitigate the curse of dimensionality and preserve model interpretability, we applied feature selection. Since embedding methods lose the physical interpretability of the features, we used a cherry-picking method. First, we reduced redundancy by pruning each pair of highly correlated features. Next, we scored the remaining candidates using a composite ranking derived from Pearson correlation, ANOVA test, and Kruskal–Wallis test, and selected the four most discriminative features (see Fig. 3). Introducing the extracted features reduces the regression test loss from 0.86 to 0.22 and the classification test loss from -0.79 to -0.85. Finally, we excluded the *np* variable since it exhibits a strong linear dependency with *ng*, and the environmental variables *ias* and *pa* after observing that the two KANs found that these features are not very relevant (a posteriori analysis).

3. METHODOLOGY

This section describes how KANs were used to predict the failure state of the helicopter turbine engine.

3.1 Probabilistic Regression of Torque Margin

Prior to classifying the engine health, the Probability Density Function (PDF) of the torque margin is predicted. Torque margin trq_{mrg} is defined as the deviation of the measured torque trq_{mes} from its target value trq_{tar}, and can be expressed as:

$$trq_{mrg} = 100 \times \frac{trq_{msr} - trq_{tar}}{trq_{tar}}. \tag{1}$$

Following (Romano et al., 2024), we restrict our analysis to a Gaussian PDF. This simplification is warranted by the Central Limit Theorem, under the reasonable assumption that the torque margin originates from the aggregation of many independent and identically distributed (i.i.d.) error sources.

We adopt a KAN architecture with layer dimensions [8, 1, 2], and to further improve interpretability with smoother

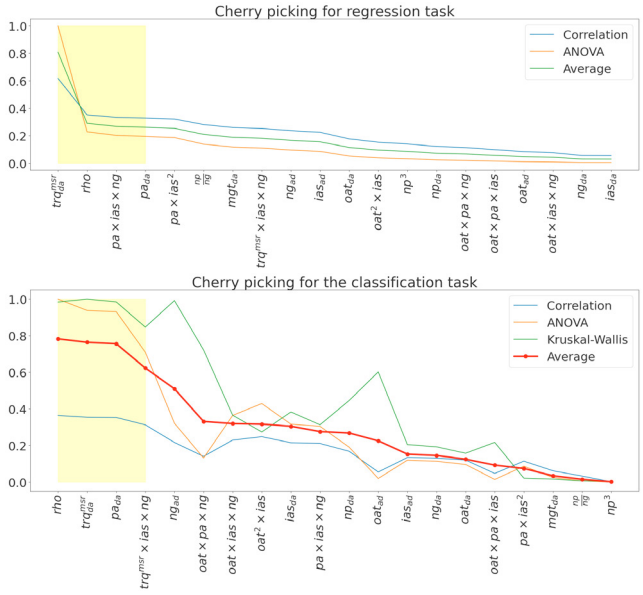

Fig. 3. The result of the cherry-picking ranking for regression (upper figure) and classification (bottom figure). The yellow regions indicate the selected features.

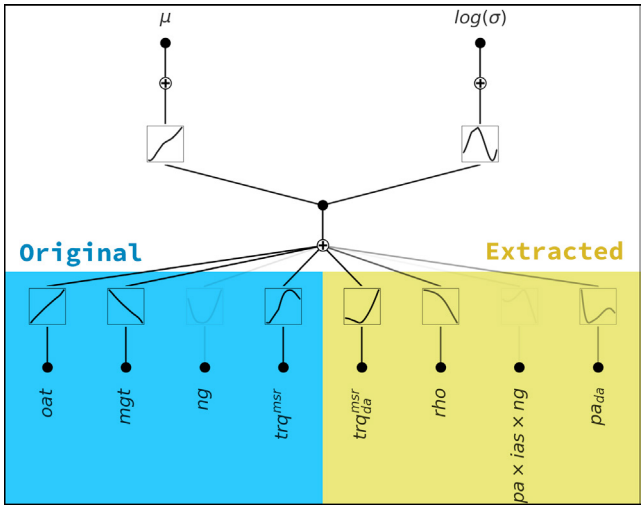

Fig. 4. The KAN trained to predict the Gaussian PDF of the torque margin. Edges opacity reflects the learned multiplicative scale weights, with more transparent splines highlighting less influent nodes.

splines, the size of the spline grid has been chosen to be small and set to 2. The two outputs parametrise the Gaussian PDF by predicting its mean and the log-variance. Predicting the logarithmic variance instead of the variance ensures that the predicted variance is always positive and improves numerical stability during training (Kendall and Gal, 2017). Furthermore, we optimized the Gaussian negative log-likelihood loss instead of the conventional mean squared error. Training was performed for 300 epochs using the first-order Adam optimizer with an initial learning rate of 0.05, an exponential decay factor of 0.96, and a batch size of 2048. Under these settings, the network converged to the configuration shown in Fig. 4.

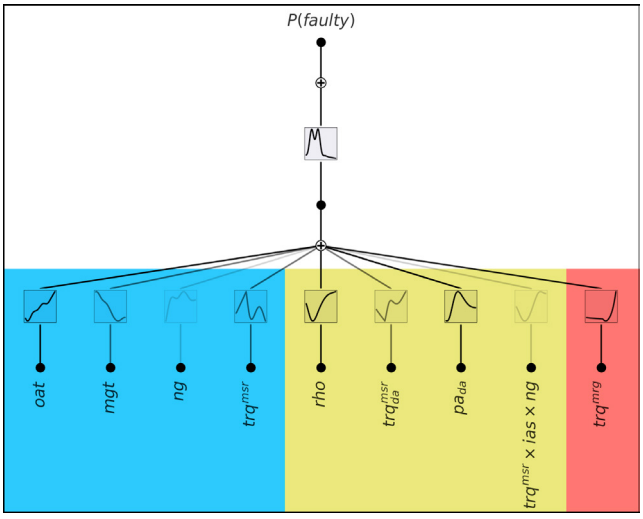

Fig. 5. The KAN trained to predict the failure state of a helicopter turbine engine. The red region indicates the torque margin predicted by the regression model.

3.2 Classification of Failure State

The mean values of the predicted torque margin PDFs were concatenated with the four base and four extracted features described in Section 2.4 to form the overall train set for the classification KAN. Similiarly to the regression KAN, we adopted a [9,1,1] architecture and trained it using the same hyperparameter settings. The only alteration concerned the loss function: instead of standard binary cross-entropy, we optimized the negative of the challenge's scoring metric. Figure 5 illustrates the final KAN configuration.

4. EXPERIMENTS AND RESULTS

The predictions from both KANs were evaluated on the test dataset by using the PHM scoring system. The test score of 0.81 (in a range between 0 and 1) confirms the effectiveness of the proposed solution in meeting the challenge and the use of KANs in data-driven predictive maintenance tasks. In this section, we focus on the two main advantages of KANs over alternative methods, namely *interpretability* and *robustness to catastrophic forgetting*.

4.1 Interpretability Trade-off

In fault detection applications, interpretability often takes precedence over accuracy. Ideally, white-box, physics-based process models are preferred for their inherent transparency; however, such models are rarely available in practice. Interpretability becomes particularly important in safety-critical domains, such as helicopter flight monitoring, as posed by the PHM challenge. In these scenarios, it is essential to understand the rationale behind each prediction. Although the considered challenge required reporting a confidence score rather than a hard class label – which, in principle, provides some degree of interpretability – MLPs still lack transparency. Even relatively small networks make it difficult to trace the influence of individual inputs on the final output.

KANs are inherently more interpretable than traditional MLPs. This stems from the fact that their logical units – activation functions placed on the edges – are significantly more expressive than those of MLPs, which rely solely on a scalar product followed by a bias (Liu et al., 2024).

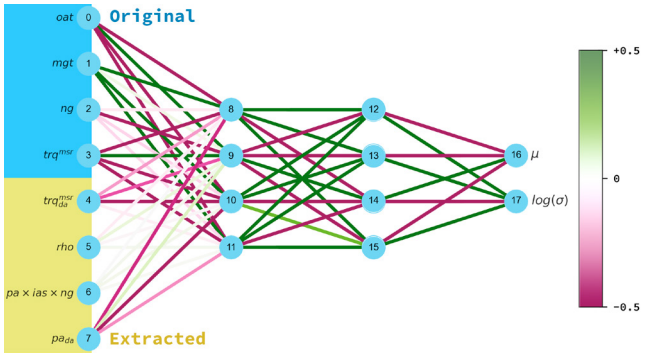

Fig. 6. MLP trained to solve the probability regression task of the PHM challenge.

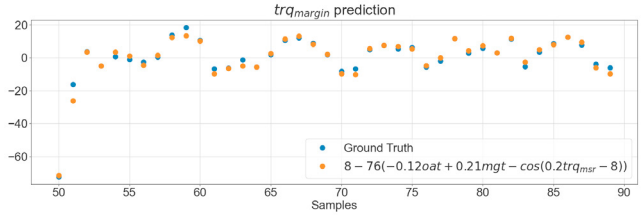

Fig. 7. The performance of the approximate closed formula derived by fixing the KAN's splines with symbolic functions.

To support our claim, Fig. 6 shows a plot of an MLP trained on the same regression task as the KAN in Fig. 4. We chose the MLP's architecture ([8, 4, 4, 1]) with the `ptflops` library to match the KAN's parameter count. While the MLP's internal mapping remains difficult to read, the KAN's spline-based activations make it clear how each feature influences the output.

This interpretability gain comes at the cost of lower predictive accuracy: the KAN in Figure 4 converges to a test Gaussian negative log-likelihood of -0.024, whereas the MLP in Fig. 6 attains a lower value of -1.196.

Another key advantage of KANs lies in their capacity for symbolic representation: each spline can be expressed analytically, enabling the derivation of approximate closed-form expressions for the model's outputs (Liu et al., 2024). To illustrate this property, we retrain the regression KAN using a reduced input set – specifically, *oat*, *mgt*, *ng*, and trq_{msr} - and following the `auto_symbolic` function from PyKAN. After a minor manual correction of improperly fixed splines, we obtain the following approximate expression for the mean of the toque margin:

$$\mu_{trq_{mrg}} \approx 8 - 76(-0.1oat + 0.2mgt - cos(0.2trq_{msr} - 8)). \quad (2)$$

As shown in Fig. 7, this simple non-linear formula yields a quite accurate approximation of the true output, proving KANs' potential for interpretable symbolic regression.

4.2 Incremental Domain Learning

In a real-world scenario, an intelligent system needs to incrementally acquire, update, accumulate, and exploit

knowledge throughout its lifetime. This ability, known as continual learning, provides a foundation for AI systems to develop themselves adaptively. This is common in real-time systems, where samples from a given domain may never recur, making it essential for the network to retain previously acquired knowledge.

(Liu et al., 2024) argue that KANs are inherently suited for CL, thanks to the *local plasticity* property of their spline-based activations. During backpropagation, updating a control point alters only a localized segment of the spline, with the affected region determined by the spline degree k. Therefore, when the KAN encounters new data from a different domain, only a subset of control points on each spline is modified, thereby preserving the previously learned configuration (Cacciatore et al., 2024).

(van de Ven and Tolias, 2019) introduce three Incremental Learning (IL) scenarios of increasing difficulty:

(1) Task-IL: At test time the model is given the task identifier and may deploy task-specific modules, making this the simplest scenario.
(2) Domain-IL: Task IDs are hidden, but the label space remains fixed while the input distribution shifts across domains. The model must adapt to new domains without explicit task recognition.
(3) Class-IL: The model must infer the current task ID and discriminate among all classes – including those never seen together – making this the most challenging scenario.

In the proposed experiment, we investigate a Domain-IL scenario to evaluate the resilience of KANs to catastrophic forgetting in comparison to MLPs. To this end, both models are trained on the classification task once again, with the dataset partitioned into two subsets: the first containing samples where $np > ng$, and the second where $np < ng$ (see Fig. 1). The test and training sets are obtained using the holdout method. During training, the networks are initially trained on the first set for a certain number of epochs, and then the second one is shown. We then repeat the experiment with replay, meaning that the two-stage training cycle is repeated multiple times, allowing each network to revisit earlier data and thereby assess its capacity to retain and reinforce previously learned knowledge.

Prior to performance comparison, we tune each architecture's shape by selecting appropriate hyperparameters. A KAN is characterized by its layer-wise node counts, the spline grid size G, and the spline degree k. In this study, we employ Blealtan's EfficientKAN [2], a streamlined and fast implementation that preserves the continual-learning performance of the original PyKAN framework when both spline parameters and SiLU multiplicative weights are frozen (Cacciatore et al., 2024).

Focusing on continual-learning performance rather than interpretability, we adopt the same moderately complex architecture with node configuration [6,256,256,1] for MLP chosen by (Romano et al., 2024) ($\approx 67.84K$ parameters). We then empirically determine $G = 20$ by evaluating test-score trends across several G values (Fig. 8), observing that performance improves with increasing grid resolution

[2] `https://github.com/Blealtan/efficient-kan`

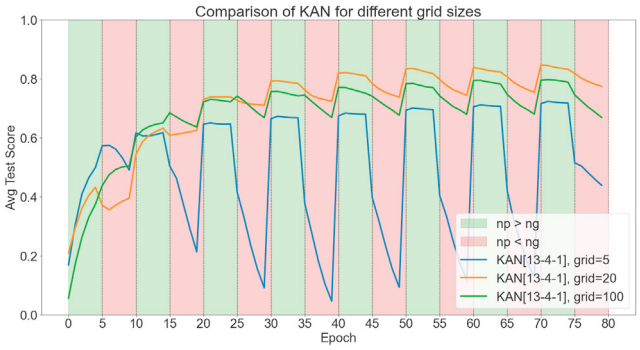

Fig. 8. Test-score trajectories under a Domain-IL protocol for three KAN configurations with different spline grid sizes. Results indicate that increasing control-point density enhances knowledge retention up to an optimal grid resolution, after which overparameterization leads to performance decline.

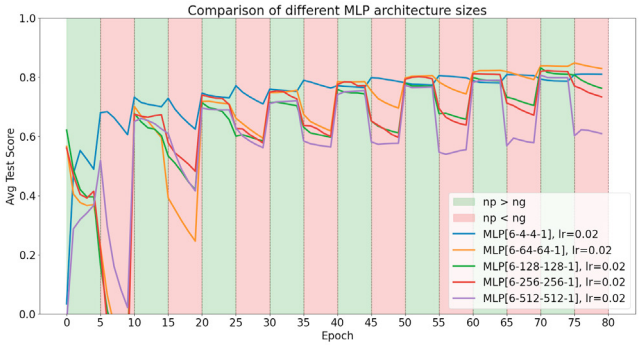

Fig. 9. Test-score trajectories under a Domain-IL protocol for multiple MLP architectures with varying layer sizes. The results demonstrate that increasing network capacity adversely affects continual-learning performance.

up to a threshold. The choice of such values is based on the authors' continual learning experiments (Liu et al., 2024).

By contrast, MLP resilience to catastrophic forgetting steadily declines as network depth and width increases (Fig. 9), indicating that – aside from very small architectures – MLPs are ill-suited to continual-learning scenarios.

We find an equivalent KAN architecture in terms of learnable parameters, using the `ptflops` library like in Sec. 4.1, arriving at [6,51,51,1] ($\approx 68.03K$ parameters). For all experiments, we trained each model for 20 epochs per domain with an initial learning rate of 2×10^{-2} and an exponential decay factor of 0.96. In the replay protocol, each domain was revisited for 5 epochs, and the two-stage cycle was repeated eight times to evaluate the models' ability to consolidate and recall prior knowledge (see Fig. 10).

In terms of computational complexity, EfficientKAN delivers a $5-6\times$ speedup over PyKAN (Cacciatore et al., 2024). However, despite operating with a similar FLOP count to the MLP ($\approx 142K$ vs. $\approx 136K$), the chosen EfficientKAN configuration requires almost ten times longer to train

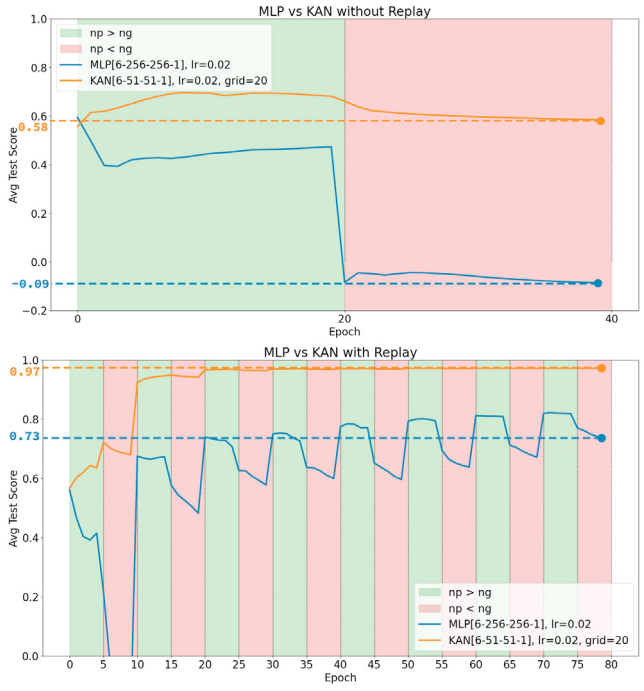

Fig. 10. Test-score trajectories for MLP and KAN models under a Domain-IL protocol, without (top) and with (bottom) replay. In both settings, KANs consistently outperform MLPs in maintaining performance across sequential tasks, demonstrating superior resilience to catastrophic forgetting.

than the MLP, due to the increased complexity of the operations performed during each forward pass.

5. CONCLUSION

This paper establishes KAN as a valuable alternative to conventional deep learning architectures for safety-critical fault detection. By construction, KANs produce inherently interpretable models whose edge-wise functions can be directly inspected. Furthermore, we show how the learned splines can be symbolically approximated to produce a closed-form formula for the outputs, thereby revealing meaningful physical relationships among the input variables.

In addition, KANs exhibit remarkable intrinsic robustness to catastrophic forgetting under incremental domain learning, making them well suited for on-line predictive maintenance pipelines where environmental conditions and engine behavior evolve over time. Their use can significantly reduce the need for corrective actions such as replaying data from previous domains.

Although our simplified KAN architecture already achieves strong performance in terms of interpretability and continual learning, future work will investigate the use of multiplicative nodes to improve predictive accuracy while preserving the network's interpretability and resilience to forgetting.

REFERENCES

Akinwande, O., Dikko, H., and Agboola, S. (2015). Variance inflation factor: As a condition for the inclusion of suppressor variable(s) in regression analysis. *Open Journal of Statistics*, 05, 754–767. doi:10.4236/ojs.2015.57075.

Alves Ribeiro, V.H. and Reynoso-Meza, G. (2024). A design science approach comparing ensemble learning and artificial neural networks for uncertainty-aware helicopter turbine engines health monitoring. In *Annual Conference of the PHM Society*, volume 16. doi:10.36001/phmconf.2024.v16i1.4187.

Cacciatore, A., Morelli, V., Paganica, F., Frontoni, E., Migliorelli, L., and Berardini, D. (2024). A preliminary study on continual learning in computer vision using kolmogorov-arnold networks. doi:10.48550/arXiv.2409.13550.

Gohil, V., Dev, S., Upasani, G., Lo, D., Ranganathan, P., and Delimitrou, C. (2024). The importance of generalizability in machine learning for systems. *IEEE Computer Architecture Letters*, 23(1), 95–98. doi:10.1109/LCA.2024.3384449.

Han, P., Liang, Q., Vanem, E., Knutsen, K.E., and Zhang, H. (2024). Assessing helicopter turbine engine health: A simple yet robust probabilistic approach. In *Annual Conference of the PHM Society*, volume 16. doi:10.36001/phmconf.2024.v16i1.4186.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5), 359–366.

Hun Park, Y., Hwan In Oh, H., Tae Kim, I., Jung Lee, S., Hee Moon, S., Jin Park, G., Park, J.K., and Ha Jung, J. (2024). Intelligent helicopter turbine engine fault diagnosis using multi-head attention. In *Annual Conference of the PHM Society*, volume 16. doi:10.36001/phmconf.2024.v16i1.4193.

Kendall, A. and Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision?

Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačić, M., Hou, T.Y., and Tegmark, M. (2024). Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*.

Ozeki, K., Masuzaki, T., Shiraga, T., Wakimoto, K., and Nakamura, T. (2024). Estimating the health of turbine engine based on the relationship between torque margin and density altitude. In *Annual Conference of the PHM Society*, volume 16. doi:10.36001/phmconf.2024.v16i1.4191.

Pankaj, P., Jain, S., and Joshi, S. (2024). A comprehensive approach to fault classification of helicopter engines with adaboost ensemble model. In *Annual Conference of the PHM Society*, volume 16. doi:10.36001/phmconf.2024.v16i1.4194.

Romano, T., Siegel, N., III, S.T.W., Henn, W., and Seshadri, R. (2024). Estimating the health of helicopter turbine engines by means of regression and classification using a probabilistic neural network. In *Annual Conference of the PHM Society*. doi:10.36001/phmconf.2024.v16i1.4196.

van de Ven, G.M. and Tolias, A.S. (2019). Three scenarios for continual learning. doi:10.48550/arXiv.1904.07734.

Wu, Z., Wang, J., and Li, M. (2024). Robust health condition prediction of helicopter turboshaft engines using ensemble machine learning models. In *Annual Conference of the PHM Society*, volume 16. doi:10.36001/phmconf.2024.v16i1.4195.