Multi-class Neural Additive Models : An Interpretable Supervised Learning Method for Gearbox Degradation Detection

Charles-Maxime Gauriat, Yannick Pencolé, Pauline Ribot, Gregory Brouillet

Multi-class Neural Additive Models : An Interpretable Supervised Learning Method for Gearbox Degradation Detection

Charles-Maxime Gauriat*†, Yannick Pencolé†, Pauline Ribot† and Gregory Brouillet*
*BOSCH, Rodez, France. Email: CharlesMaxime.Gauriat@fr.bosch.com, Gregory.Brouillet@fr.bosch.com
†LAAS-CNRS, Université de Toulouse, CNRS, INSA, UPS, Toulouse, France. Email: ypencole@laas.fr, pribot@laas.fr

***Abstract*—This paper introduces the Multi-class Neural Additive Models (MNAMs), an extension of Neural Additive Models (NAMs) designed to solve multi-class classification problems while remaining interpretable. The paper firstly presents a set of definitions around model interpretability and associated concepts, concepts on which the proposed machine learning method relies on. The core of the contribution lies in the development of MNAM, a model training method designed to minimize the traditional trade-off between accuracy and interpretability. This method is then put to test in a concrete application: the detection of gearbox degradation levels using vibration data, as part of the PHM Society data challenge of 2023. The obtained results demonstrate that a MNAM reaches higher accuracy performance than other interpretable methods such as Decisional Tree (DT) or Generalized Additive Models (GAMs).**

I. Introduction

In the industrial field and for the last decade, factories are more and more seeking to move away from corrective maintenance to condition based maintenance (CBM) [1]. By instrumenting complex systems in order to acquire data, they can develop a data-driven approach [2] able to compel with a Prognostic and Health Management (PHM) strategy and therefore moving toward predictive maintenance. Predictive maintenance consists in optimally deciding when to replace a component in a complex system like a machine tool so that it is always operating properly [3]. It also helps to prevent manufacturing waste.

To get such a predictive maintenance strategy, relevant sensors are added to components of the complex system to measure data at operating time and use a prognostic model to check the current health and predict the Remaining Useful Life (RUL) of every component of the machine at any time. Through its life, a component will reach various degradation levels until it is no longer functional. With a data-driven approach, it is possible to detect and predict when these degradation levels will appear. Various data processing techniques and machine learning (ML) methods are used to determine the RUL [4]. However, as performance increases, confidence in these algorithms can decrease. As in the medical and legal fields, the industry needs more than ever to be able to better understand the ML methods that are used and the decision-making process of the trained models, in order to avoid the use of so-called black box models [5]. With a better understanding of the trained models, data scientist experts can indeed more effectively detect deviations and learning bias. However, in a PHM context, maintenance operators must also have a better understanding of the decisions produced by the trained models, as the operators are responsible for the efficient maintenance of their equipment. To ensure that operators follow the predictive maintenance strategy obtained by machine learning techniques, the underlying trained model must offer an acceptable level of confidence. This can only be achieved through the use of interpretable models.

There are plenty of supervised learning methods available at different scales that aim at learning interpretable models such as Linear Models, Decision Trees (DT) or Generalized Additive Models (GAMs) [6]. However, more complex and performant methods such as Multi Layer Perceptron (MPL) are still considered as black box models so far, as the interactions between the hidden layers of the model cannot be interpreted. Recently, to overcome this issue in such models, Neural Additive Networks (NAMs) [7] have been introduced. This supervised method proposes to use the concepts of GAMs applied to neural network structures. To date, NAMs have been able to solve supervised task like regression problems or binary classification problems [8]. However, to solve some PHM problems, multi-class classification is required: for instance in the classification of fault degradations or ageing models that characterize a RUL as a set of classes [9].

The contributions of this paper are the following. First, model interpretability and related concepts are discussed leading to a fixed set of definitions that will be used throughout this paper. The second contribution is the proposition of the Multi-class Neural Additive Model (MNAM) as an extended version of the NAM algorithm for multi-class classification to train models that are interpretable. A MNAM is developed to reduce the accuracy-interpretability tradeoff that can be found with other interpretable methods. With help of this new model, a new supervised learning method is presented that learns models for detecting degradation levels. To compare the proposed method and illustrate its advantage with regards to other supervised learning techniques, MNAMs and these methods have all been implemented and tested on the PHM

Society Data Challenge 2023 [10] which is a problem of multi-class degradation detection relying on gearbox vibratory data.

The paper is organized as follows. Section II discusses the different notions about interpretability. Section III presents the MNAM architecture. In Section IV, the proposed MNAM used for learning degradation models is applied to the PHM Data challenge and its performance is experimentally compared with other methods. The interpretation of the obtained model is then detailed. Section V finally discusses the performance and the limitations of the MNAM.

II. About interpretable models

Lately, the Explainable Artificial Intelligence (XAI) community has been using various terms referring to the comprehension of machine learning models: interpretability, explainability, intelligibility or even comprehensibility [11]. As the vision behind these concepts seems to be fuzzy and does not refer to a monolithic concept so far [12], it is decided for this paper to propose the following definitions.

Explainability is the ability to explain a prediction. It is a post prediction process where the model is able to justify *the* prediction using the feature interactions that the model has used to perform this specific prediction.

Interpretability is a stronger concept than explainability (interpretability $\Rightarrow$ explainability). An interpretable model is able to give overall feature-based rules as functions that map a feature to *every* possible future prediction.

Unlike explainable models, it is not related to a post prediction process and it does not depend on the current prediction of the model. Whatever the set of predictions the model will perform in the future, they can be explained by these feature-based rules. This concept ensures the algorithmic transparency: any user is able to make a prediction by using these rules so an interpretation by the user of any prediction is possible.

The last concept is *intelligibility* that can be associated with explainability and interpretability. This is about the ability of the model to be understood by a human. It is achieved by keeping the explanation as minimalist as possible. Lesser are the number of features and rules used to make a prediction, and higher will be the intelligibility. Intelligibility is bringing up together concepts as simulatability, decomposability from [12] and comprehensibility from [11] which are about maximizing the human comprehension using the minimum set of rules.

Known supervised learning methods are more or less explainable or interpretable. For example, explainability is available in DT, as it is possible to explain a prediction in terms of the features involved in the prediction. This can be obtained with a feature importance plot for this prediction. But actually, DT is interpretable. A DT rule is defined by a branch of the tree and the thresholds defined in any node of the branch. All rules can be displayed, and a human can manually make the prediction just following those rules. This is why DT is considered to be interpretable. However, it is the shortness of the tree that makes it intelligible. The fewer nodes needed to make a prediction, the greater the intelligibility.

GAMs are also explainable using an additive structure, which allows each variable to be treated separately. The impact of each variable on a prediction is thus visible. GAM interpretabilty lies in the rules given by splines that capture the linear and non-linear relations between the input and output features. Intelligibility depends on the number of involved features and how easy they are to read.

In Multi Layer Perceptron (MLP), explainability can be achieved by observing the feature importance plot for a given prediction, but also with agnostic tools like SHapley Additive exPlanation (SHAP) [13] that gives a detailed explanation of the output of any model. However, precise model rules are not available, as a fully connected network is very complex. Next section introduces MNAM, which takes advantage of the training performance of neural networks like MLP, while the trained models are interpretable.

III. Multi-class Neural Additive Models

This section focuses on the definition and the use of a Multi-class Neural Additive Model (MNAM), which is an interpretable supervised learning method for multi-class classification problems. The structure of Neural Additive Models (NAM) is first briefly summarized. Next, the structure of MNAM is introduced, with its differences from the original NAM. Finally, the way in which trained MNAM can be interpreted is detailed.

A. Supervised learning problem: mathematical notations

In order to understand the structure of these methods, here are provided some mathematical definitions that will be used throughout the paper. Let A be the set of features, such that $A = \{a_1, \ldots, a_{|A|}\}$. The domain of a feature a_i is $\mathcal{X}_i$. Let $\mathcal{X}$ be the set of all possible individuals defined by the features of A. $\mathbf{x} \in \mathcal{X}$ is an individual such as $\mathbf{x} = [x_1, \ldots, x_{|A|}]$ with $\forall i \in \{1, \ldots, |A|\}, x_i \in \mathcal{X}_i$. A *dataset*, denoted X, is the available set of N individuals for training, testing and validating the trained models. So, $X \in \mathcal{X}^N$ is the data matrix of dimension $N \times |A|$ that will be used as a set of inputs to the models, such that:

$$X = \begin{bmatrix} \mathbf{x_1} \\ \vdots \\ \mathbf{x_N} \end{bmatrix} = [\mathbf{a}_1, \ldots, \mathbf{a}_{|A|}] = \begin{bmatrix} x_{1,1} & \cdots & x_{1,|A|} \\ \vdots & & \vdots \\ x_{N,1} & \cdots & x_{N,|A|} \end{bmatrix}$$

with $\mathbf{a}_i \in \mathcal{X}_i$ the vector composed by N values for the feature a_i such as : $\mathbf{a}_i = \begin{bmatrix} x_{1,i} \\ \vdots \\ x_{N,i} \end{bmatrix}$

This paper deals with multi-class classification, and C denotes the number of classes that are named by the indexes $\{1, \ldots, C\}$. Any individual in the dataset X is associated with one of these classes. The aim of any supervised learning

method is then to produce a model that predicts a class c given an input $\mathbf{x} \in \mathcal{X}$ with $c \in \{1, \ldots, C\}$.

B. Neural Additive Models

Neural Additive Models (NAMs) are glass-box models defined in [7] which use a methodology belonging to the family of Generalized Additive Models [14] known for their ability to capture linear and non-linear relations between predictive features and predictions while remaining interpretable. Its strength lies in using the versatility of Deep Neural Networks (DNNs) instead of boosted trees.

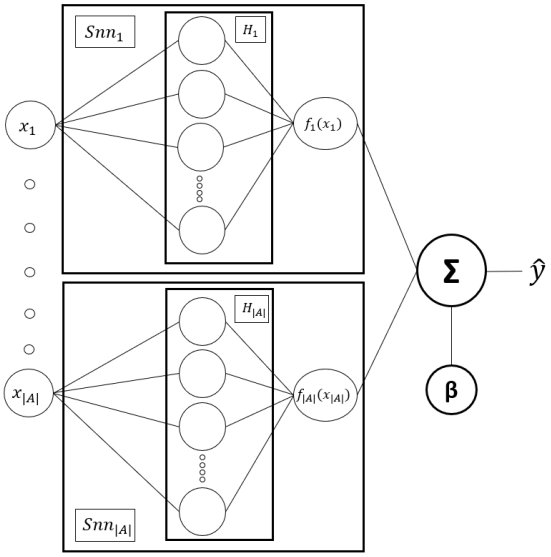

Fig. 1. NAM architecture

The particularity of a NAM is that it is composed of a set of shallow Neural Networks, one network for each feature a_i (called the feature network S_{nn_i}) and a structure that combines their outputs to perform the predictions $\hat{y}$. The predicted output $\hat{y}$ of the NAM for an individual $\mathbf{x}$ is defined as follows :

$$\hat{y} = \sum_{i=1}^{|A|} f_i(x_i) + \beta \tag{1}$$

where x_i is the value of $\mathbf{x}$ for the feature a_i. $f_i(x_i)$ is the feature network output associated with the feature a_i and β is the bias parameter. In Figure 1, each feature network S_{nn_i} is composed of its input x_i, a structure H_i made up of successive hidden layers defined during the model design phase, and its output $f_i(x_i)$. The H_i structure can be composed of several hidden layers, generally made up of regular units, using a ReLU activation function.

One of the problems with S_{nn} that have only one input feature is that they often struggle to approximate 1D sharp jump functions with the regular unit and a ReLU activation function on the first layer of H_i. This is why the authors in [7] propose the use of a new hidden unit named EXp-centered-Unit (ExU) that can learn and adjust the weight parameters in logarithmic space. Each new hidden unit using an activation function σ compute $h(x)$ given by :

$$h(x) = \sigma(e^w(x - b)) \tag{2}$$

with x the input value of the hidden unit, w the weight parameter and b the shifting bias. In this way, it can be used to overfit 1D sharp jump functions. This hidden unit is preferably placed in the first layer of the H_i structure.

C. Multi-class Neural Additive Models

NAMs are efficient when dealing with regression and binary classification problems. However, they are not designed for multi-class classification. We propose Multi-class Neural Additive Models (MNAMs) as an extension of NAMs.

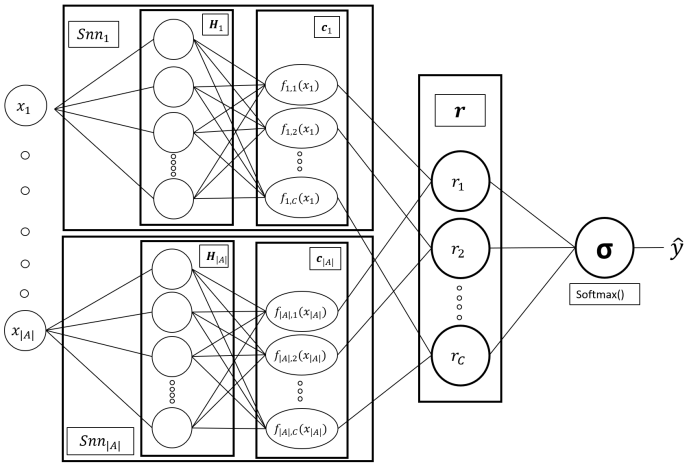

Fig. 2. MNAM architecture

1) MNAM structure: The structure of a NAM does not allow for multi-class classification, as the output is composed of a single neuron. With a classical MLP, there are two ways of performing multi-class classification. The first method is the one-vs-all model, in which several models learn a binary classification between one class and the others. The second method uses as many outputs as there are classes. The latter method is chosen to keep only one model training, with all parameters updated at the same time. Like the model output, the structure of each S_{nn} is modified to comply with the multi-class classification. In the multi-class context, S_{nn_i} must produce an output for all available classes. To achieve this, the structure of S_{nn_i} is modified as shown in Figure 2 to replace the output $f_i(x_i)$ by a vector $\mathbf{c_i}$ defined as follows:

$$\mathbf{c}_i(x_i) = \begin{bmatrix} f_{i,1}(x_i) \\ \vdots \\ f_{i,C}(x_i) \end{bmatrix} \tag{3}$$

where $f_{i,j}(x_i)$ is the output by S_{nn_i} of the input x_i associated with class $j \in \{1, \ldots, C\}$. Next, the sum of all the outputs of the feature networks is replaced by a vector $\boldsymbol{r}$

composed of the sums of all feature network outputs for each class.

$$\boldsymbol{r}(\mathbf{x}) = \begin{bmatrix} r_1(\mathbf{x}) \\ \vdots \\ r_C(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{|A|} f_{i,1}(x_i) + \beta_1 \\ \vdots \\ \sum_{i=1}^{|A|} f_{i,C}(x_i) + \beta_C \end{bmatrix} \tag{4}$$

with β_j the bias with $j \in \{1, \ldots, C\}$. Since the network now has C outputs, a softmax function is applied to transform $\boldsymbol{r}$ into a probability distribution that produces the MNAM output $\hat{\mathbf{y}}$:

$$\hat{\mathbf{y}} = \sigma_{softmax}(\boldsymbol{r}(\mathbf{x})) = \left[\frac{e^{r_j(\mathbf{x})}}{\sum_{j=1}^{C} e^{r_j(\mathbf{x})}} \right]_{j \in \{1,\ldots,C\}} \tag{5}$$

As with NAM methods, ExU can be used on the first layer of each S_{nn} of MNAM methods. However, this unit cannot learn sharp function when x is negative. This problem is solved by the ExpDive hidden unit proposed by [15]. ExpDive hidden unit is defined by:

$$h(x) = \sigma((e^{w} - e^{-w})(x - b)) \tag{6}$$

with x the input value, w the weight parameter, b the shifting bias and σ the activation function of the ExpDive hidden unit.

2) MNAM interpretability: As with NAMs and GAMs, interpretability is based on two tools : feature importance and shape function plots. A feature importance plot measures the average impact of all features on the final prediction score. These indicators provide information on the rules learned by all S_{nn}. Feature importance of MNAM methods is obtained by determining the mean absolute score I_i of S_{nn_i} based on dataset X and is defined by:

$$I_i = \frac{1}{NC} \sum_{j=1}^{C} \sum_{k=1}^{N} \left| f_{i,j}(x_{k,i}) - \frac{1}{NC} \left(\sum_{p=1}^{C} \sum_{m=1}^{N} f_{i,p}(x_{m,i}) \right) \right| \tag{7}$$

By plotting $I_i, i \in \{1, \ldots, |A|\}$, we can see which feature has the greatest impact on the prediction. The use of this indicator can be helpful to perform feature selection in order to reduce the number of dimensions. It can also be used to detect features that introduce bias during model learning.

Shape functions are, however, the exact description of the model decision process for all features [7]. The shape function of feature a_i for the class $j \in \{1, \ldots, C\}$ is given by the plot of all predictions from dataset X:

$$\{(v_i, f_{i,j}(v_i)), \forall v_i \in \{x_{k,i}, k \in \{1, \ldots, N\}\}\}. \tag{8}$$

where the expression $v_i \in \{x_{k,i}, k \in \{1, \ldots, N\}\}$ corresponds to the set of single values of the vector $\mathbf{a}_i$ of feature a_i. By using shape functions, users get full transparency on the model rules and are able to perform the prediction manually, as all predictions from a class only have to be summed to make the prediction. In addition, the density of individuals can be displayed using shape functions. This provides information on the distribution of individuals in the dataset. This can be used to detect potential biases, as well as a lack of model robustness on under-represented data.

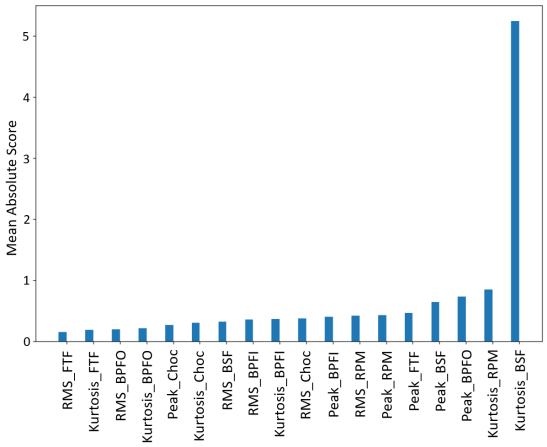

Fig. 3. Basic dataset feature importance

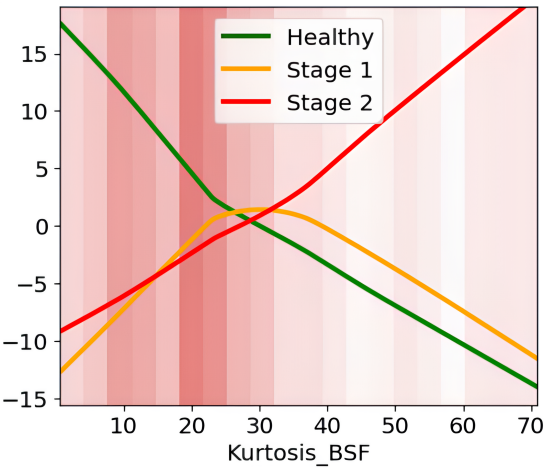

Fig. 4. Shape functions of the Kurtosis_BSF feature

To illustrate how interpretability works in a MNAM, a basic dataset produced by Bosch Rodez industry is used as an example. This dataset is composed of amplitude values for specific vibration frequencies of spindle bearings in operation (like the specific BPFO, BPFI frequencies for bearings [16]). The data have been recorded for several months and capture several run-to-failure signals of spindle bearings rotating at 90K rotation per minute and have been labeled from healthy stage to the most degraded stage 2. The MNAM is trained on the original measured frequencies and on some inferred statistics like RMS (Root Mean Square) and Kurtosis. After training, the feature importance plot presented in Figure 3 suggests that the kurtosis of the ball (BSF) is the input feature with the greatest impact on the prediction. The shape functions

for this feature presented in Figure 4 clearly define a rule between the BSF-kurtosis and the degradation: for a BSF-kurtosis within the range [0, 25], $S_{nn}(BSF_{Kurtosis})$ predicts a healthy stage, for a range in [25,32] the stage 1 and for [32,72] only stage 2. In this example, this rule is simple enough for a human to understand the model's decision and is therefore intelligible.

Based on this type of rules, the shape function of feature a_i on class j provides a direct relation between feature a_i and the prediction of model S_{nn_i} with respect to class j.

To summarize, thanks to the feature importance plot and the shape function plot, operators are able to understand the model's reasoning. Moreover, they can also perform a manual prediction and obtain the same results as the model, hence the interpretability of the model.

IV. Application to a problem of degradation detection

This section presents how MNAM methods can be applied to learn models for the detection of degradations and a comparative analysis with a set of classical supervised learning techniques. To perform the comparison, the dataset from the data challenge PHM2023 is used. To evaluate the method, a benchmark is set up to compare MNAM accuracy and loss performance with respect to these other methods. The interpretation provided by the trained MNAM for this challenge is finally presented.

A. Data challenge PHM2023

The data challenge that is used for these experiments has been proposed by the Prognostic and Health Management (PHM) Society in 2023 [10].

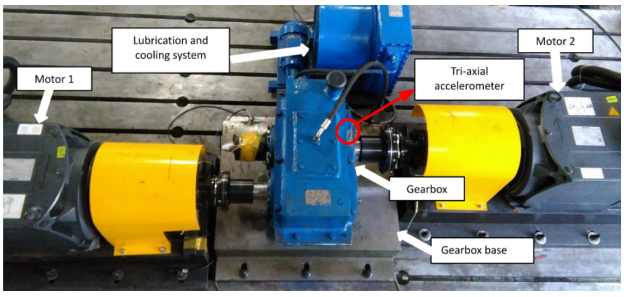

Fig. 5. Data challenge test rig [10]

The purpose of this challenge is to develop an estimate of the severity of a fault in a gearbox, where each class represents a level of fault. To simulate this degradation, an experimental test rig has been set up with two motors and a gearbox. The first motor serves as a drive motor, while the other acts as the load motor (see Figure 5). To measure the fault level of the gearbox, a triaxial accelerometer has been fixed to the gearbox case, close to the bearing house. Vibration data were collected at 20,480 Hz for the horizontal, vertical and axial axes under several conditions of rotation speed and torque. The test rig recorded 11 fault levels, from 0 (healthy) to 10 (most damaged).

The original dataset is divided into three parts for the purpose of the challenge: a training set, a test set and a validation set. The training set contains 7 fault levels. The other two sets are composed of the same levels, but also contain the 4 missing levels. This paper uses only the training set, as labels for the other sets are currently unavailable.

B. Data processing

The original dataset consists of acceleration records for 3 axes measured in m/s^2 (horizontal, axial and vertical). Measured velocity and torque, respectively measured in rotations per minutes (RPM) and Newton (N), are given for all records. For each axis, the condition record is split up into a vector s of signals with 20480 values, depending on the test rig recording frequency.

It is common to use Fast Fourier Transform (FFT) for analyzing the frequency components of the signal. However, when dealing with high frequency signals, the FFT yields a high-dimensional output encapsulating a large range of features corresponding to the various frequency components. A high number of feature means an increase of the complexity of the MNAM which could lead to a curse of dimensionality [17], [18]. Another risk is a loss of intelligibility for the MNAM, as the set of rules would increase. This is why we propose to use statistic measures to capture the trends and derivations of all signals. For each signal on the three axes, three statistical features are computed, hence the production of 9 new features. First, the Root Mean Square (RMS) value of an acceleration signal vector s is defined by :

$$RMS(s) = \sqrt{\frac{1}{G}\sum_{i=1}^{G} s_i^2} \tag{9}$$

with $s_i \in s$ a value of the signal s and G the length of s. Secondly, the peak of the measured signal is defined by:

$$Peak(s) = max(s_1, \ldots, s_G) \tag{10}$$

The third feature is the crest, defined as follows:

$$Crest(s) = \frac{Peak(s)}{RMS(s)} \tag{11}$$

Each new signal feature is concatenated into individual vectors which are completed by the rotation speed value, the torque. The processed dataset consists of $|A| = 11$ features, $N = 12555$ total of individuals and completed by a label vector composed of $C = 7$ classes that are the pitting degradation levels. All values $x_{i,j}$ of the individuals $\mathbf{x}_i$ are normalized between 0 and 1.

The new dataset is mixed and divided into a training set (80%) and a test set (20%). An evaluation is now set up to compare different learning methods in terms of their performance and interpretability levels.

C. Models training and benchmarks

1) Models: Several supervised Machine Learning (ML) methods have been tested on this dataset. The choice is to compare the MNAM with other classical methods. Some of them are highly accurate (namely XGBoost (XGB) and Multi Layer Perceptron (MLP)) but are black-box while others are more interpretable but less accurate (namely Decisional Tree (DT) and Generalized Additive Models (GAM)). All methods, except GAM and MNAM, are trained with a Gridsearch strategy to automatically adjust the model structure and its hyperparameters in order to optimize their respective performance.

Some of the most important tuned hyperparameters are the maximum depth (DT) and the number of tree estimators (XGB). Different combinations of the number of neurons per hidden layer for the MLP were also tested. About the GAM settings, as all variables of the investigated dataset are continuous, the model is set up with $|A|$ splines. The number of knots by spline is iteratively increased to maximize performance. The final model is trained with 50 knots per spline.

About the MNAM settings, the tuning process is done manually. The training phase has been set to use a batch size of 1024 individuals for 300 epochs. Categorical cross-entropy loss is chosen as it is one of the most widely used loss functions for multi-class classification problems. Stochastic gradient descent was performed by the ADAM optimizer, as it is computationally efficient and well suited for large data sets with many parameters [19]. The model consists of a first layer of ExpDive units. Two hidden layers of 256 and 128 neurons respectively, using a ReLU activation function are added to the ExpDive layer. It is important not to add more layers, as the complexity of each S_{nn} would be unsustainable.

2) Evaluation and results: Two metrics have been selected to evaluate the results of the different ML methods. The problem being a multi-class classification, Balanced Accuracy (BA) has been chosen for its robustness to the presence of unbalanced classes in a dataset [20]. The second metric is the logloss score, which is penalized if the algorithm predicts erroneous classes with a high confidence level. The lower this score, the better the model's performance. The results are presented in Table I and show the performance metric scores on the created test set (20% of the whole data).

TABLE I
DATA CHALLENGE RESULTS

metric \ method	*MNAM*	*DT*	*GAM*	*XGB*	*MLP*
BA (%)	92.03	88.13	88.11	96.32	96.11
Logloss	0.273	2.977	0.406	0.378	0.103

The results show that MNAM is able to achieve a higher BA score than DT and GAM, but still lags behind MLP and XGB. This is understandable, given that MNAM treats each feature independently. In terms of logloss score, MNAM is in second place, just behind MLP, and therefore outperforms XGB. Recall that a high logloss score is due to the algorithm's confidence in predicting the wrong class. For DT, this score explodes as all its predictions are close to 100% confidence.

By carrying out this comparative study, MNAM can be positioned in terms of its performance in comparison with other conventional, high performing methods.

D. Data Challenge interpretability with MNAM

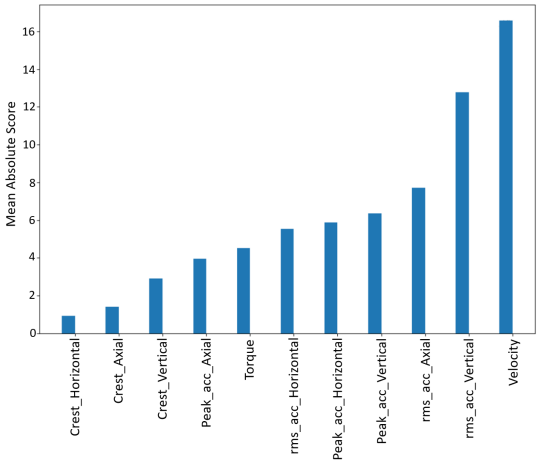

Fig. 6. MNAM feature importance

Figure 6 presents the feature importance of the trained model. It shows that velocity and vertical acceleration are the features that provide the most important information about the gearbox degradation (followed by axial and horizontal axes). RMS values seem to better capture the level of gearbox degradation than the peak and the crest values, which have little impact on the model prediction. Since velocity directly depends on the vibration level, it is understandable that this feature should have the highest importance score. However, it is interesting to note that torque has less impact than RMS acceleration and peak values for each axis.

Figures 7 and 8 present the shape function plots of the two most important features. About the velocity, Figure 7 shows a different behavior for all its shape functions. The prediction score for the degradation levels 6 and 8 (brown and black curves) seems to constantly increase with a higher RPM. This means that the probability of having one of these two classes on the output prediction of the trained MNAM increases with RPM. The curves for the levels 0 and 1 (blue and green) decrease in the [0,500] RPM range before stabilizing around a prediction score of 0. This means that their contribution to the final prediction above this velocity threshold is neutral. For the degradation levels 2 and 3 (yellow and orange curves), the score decreases as the velocity increases. This means that the probability of having one of these class for the final prediction

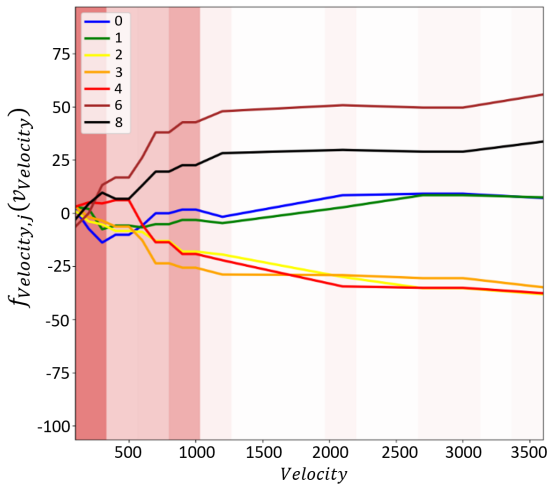

Fig. 7. Shape functions for velocity

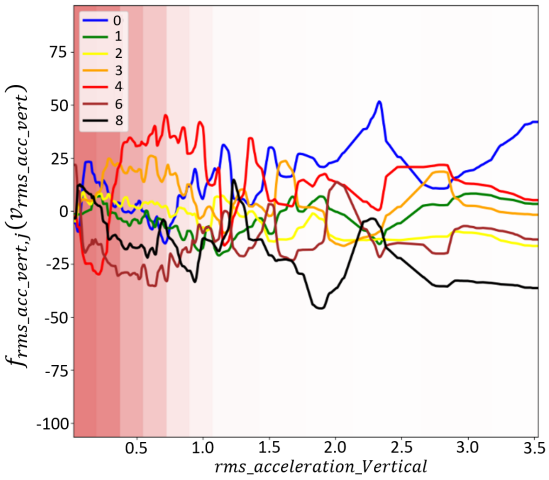

Fig. 8. Shape functions for RMS vertical acceleration

is reduced as RPM increase. The same behavior is observed for the shape function of the degradation level 4, which predicts a stable score over the [0, 700] RPM, then decreases as the velocity increases.

About the RMS acceleration, feature plot presented in Figure 8 shows that the behavior of each shape function has no clear trend, as the curves are almost sinusoidal. It is important to note that above 1.0 RMS vertical acceleration, the density of individuals is low. This reduces confidence in the model's prediction for these values. In the range [0.0,0.08], the degradation levels 6 and 8 have a higher prediction score, followed by the level 0 (healthy) in the range [0.08, 0.3] and then levels 3 and 4 in the range [0.3, 1.1], which is confusing because in a context of progressive degradation (from 0 to 8), one would expect to see a higher prediction score for most degraded levels only with higher RMS_acceleration_values values as input. In a normal case, the data recorded by an accelerometer through the lifetime of a machine tool increases as the degradation progresses. The decision-making process of the MNAM suggests that these degradation levels are learned as fault classes, not as progressive degradation.

V. Discussion

In this section, we discuss the accuracy-interpretability trade-off of the MNAM and its positioning compared to other methods as seen in Figure 9. MNAM uses the same intepretability tools as NAM or GAM, but in a multi-class context that adds more complexity. For each class j with $j \in \{1, \ldots, C\}$, interpretability is given by the sum of all $f_{i,j}(x_i)$. To perform a manual prediction using NAM, it is sufficient to use this sum. In MNAM, as only the class with the maximum score will be predicted, it is necessary to determine all r_j given an input $\mathbf{x}$, which can be complex as the number of classes increases. However, a significant advantage of MNAM methods is their ability to provide counterfactual explanations. When the model erroneously predicts a class $j = 1$ while the ground reality is $j = 2$, the use of shape functions facilitates the analysis of features influencing this incorrect decision. By identifying the specific attributes that led to a preferred class over another, MNAM enhances the transparency of the model's decision-making process.

As shown in Figure 9, MLP is considered the most black box of the algorithms used here due to its complex and fully connected structure. XGBoost employs an additive approach to make a prediction, allowing traceability in the decision-making process on the final prediction, which make this method more explainable but not interpretable yet. On the other hand, MNAM, sharing the same methods as NAMs and GAMs to achieve interpretability, is placed at an equivalent high level of interpretability. With the help of shape functions, all the rules learned by the model are explicitly given. However, while the shape functions are presented in the form of a curve, they actually represent an approximation on a scatter plot. This is why these models are considered less interpretable than a DT which has precise rules.

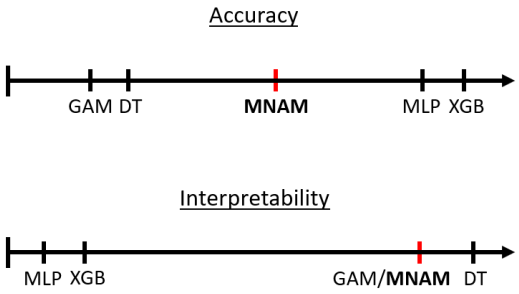

Fig. 9. Performance positioning for the data challenge

About intelligibility, the relations between independent features and the predicted class may be more complex to interpret than with a classic NAM, due to the number of shape functions per feature. In a classical NAM, by examining the evolution

of the shape function for a specific feature, one can intuitively grasp how variations in that feature influence the outcome of the model. It is even more straightforward for binary classification task as the direction of the shape function curve directly indicates its influence on the prediction (positive or negative). In multi-class classification, there is an interplay between the shape functions of all classes $f_{i,j}(x_i), j \in \{1, \ldots, C\}$ that can be visually misleading [21]. This can reduce the model's ability to be intelligible. Just as the number of features is an obstacle to understanding a model, it seems that the number of classes poses a similar challenge.

This complexity arises with the data challenge, especially in the RMS_acceleration_vertical shape function plot, where shape functions become entangled. However, it should also be noted that for the DT, the number of nodes obtained after training was 1711, which is too high to be intelligible. One reason behind this lack of intelligibility could be that the different fault levels do not accurately reflect the progressive degradation in this data processing. Another reason could be linked to the use of only the training set proposed by the data challenge, which may induce significant overfitting in the prediction. This can be observed in the shape function behavior when the density of individuals is low. Therefore, at this stage, it is preferable to use MNAM methods with fewer classes.

Nevertheless, the aim of this work is to approach performances of a MLP that fully exploit the relations between features using full-connected layers. Without these connections, MNAM succeeded in halving the performance-interpretability trade-off, with an accuracy of 92%, compared with 88% for DT and 96% for XGB on the data challenge. This positions MNAM between highly accurate but black-box methods and less accurate but interpretable methods as seen in Figure 9. Moreover, MNAM's interpretability through shape functions enables the detection of bias, potential overfitting, and makes it easier to perform feature selection.

VI. Conclusion

This paper introduces Multi-class Neural Additive Models, an adaptation of the NAMs developed by [7] designed to address multi-class classification problems while maintaining interpretability. This implementation has been tested for detecting levels of degradation in a gearbox using data from the PHM Society Data Challenge of 2023. Its performance demonstrates the possibility of reducing the trade-off between the best performing methods and the interpretable ones. Further work is currently underway to improve both the performance and intelligibility of MNAM methods. The goal remains to provide maintenance operators with the simplest understanding of the model's decision-making process, even when dealing with multiple degradation or ageing classes.

References

[1] M. Moleda, B. Małysiak-Mrozek, W. Ding, V. Sunderam, and D. Mrozek, "From corrective to predictive maintenancemdash;a review of maintenance approaches for the power industry," *Sensors*, vol. 23, no. 13, 2023.

[2] S. Yin, X. Li, H. Gao, and O. Kaynak, "Data-based techniques focused on modern industry: An overview," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 1, pp. 657–667, 2015.

[3] "Condition monitoring and diagnostics of machines — Prognostics — Part 1: General guidelines," 17.160 Vibrations, shock and vibration measurements, Standard, Sep. 2015.

[4] M. Achouch, M. Dimitrova, R. Dhouib, H. Ibrahim, M. Adda, S. Sattarpanah Karganroudi, K. Ziane, and A. Aminzadeh, "Predictive maintenance and fault monitoring enabled by machine learning: Experimental analysis of a ta-48 multistage centrifugal plant compressor," *Applied Sciences*, vol. 13, no. 3, 2023.

[5] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, pp. 206–215, 05 2019.

[6] Y. Lou, R. Caruana, and J. Gehrke, "Intelligible models for classification and regression," in *Knowledge Discovery and Data Mining*, 2012. [Online]. Available: https://api.semanticscholar.org/CorpusID:7715182

[7] R. Agarwal, L. Melnick, N. Frosst, X. Zhang, B. Lengerich, R. Caruana, and G. E. Hinton, "Neural additive models: Interpretable machine learning with neural nets," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 4699–4711.

[8] S. Moslehi, H. Mahjub, M. Farhadian, A. Soltanian, and M. Mamani, "Interpretable generalized neural additive models for mortality prediction of covid-19 hospitalized patients in hamadan, iran," *BMC Medical Research Methodology*, vol. 22, 12 2022.

[9] E. T. Chelmiah, V. I. McLoone, and D. F. Kavanagh, "Remaining useful life estimation of rotating machines through supervised learning with non-linear approaches," *Applied Sciences*, vol. 12, no. 9, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/9/4136

[10] Y. Qu, J. William, A. Saxena, N. Eklund, and S. Clements, "An introduction to 2023 PHM data challenge: The elephant in the room and an analysis of competition results," *Proc. Annu. Conf. Progn. Health Manag. Soc.*, vol. 15, no. 1, Oct. 2023.

[11] J. Marques-Silva and A. Ignatiev, "No silver bullet: interpretable ml models must be explained," *Frontiers in artificial intelligence*, vol. 6, p. 1128212, 04 2023.

[12] Z. C. Lipton, "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery." *Queue*, vol. 16, no. 3, p. 31–57, jun 2018.

[13] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 4768–4777.

[14] T. Hastie and R. Tibshirani, "Generalized Additive Models," *Statistical Science*, vol. 1, no. 3, pp. 297 – 310, 1986.

[15] M. Kim, H.-S. Choi, and J. Kim, "Higher-order neural additive models: An interpretable machine learning model with feature interactions," 09 2022.

[16] S. M.-K. Jawad and A. Jaber, "Bearings health monitoring based on frequency-domain vibration signals analysis," *Engineering and Technology Journal*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:252476550

[17] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966.

[18] Z. Zhou, J. Mo, and Y. Shi, "Data imputation and dimensionality reduction using deep learning in industrial data," in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, 2017, pp. 2329–2333.

[19] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.

[20] V. García, R. A. Mollineda, and J. S. Sánchez, "Index of balanced accuracy: A performance measure for skewed class distributions," in *Iberian Conference on Pattern Recognition and Image Analysis*, 2009. [Online]. Available: https://api.semanticscholar.org/CorpusID:1855436

[21] X. Zhang, S. Tan, P. Koch, Y. Lou, U. Chajewska, and R. Caruana, "Interpretability is harder in the multiclass setting: Axiomatic interpretability for multiclass additive models," 10 2018.