Information Fusion 99 (2023) 101882

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full length article

Exploring the balance between interpretability and performance with carefully designed constrainable Neural Additive Models

Ettore Mariotti [a,*], José María Alonso Moral [a], Albert Gatt [b]

[a] Centro Singular de Investigacion en Tecnoloxias Intelixentes (CiTIUS), Universidade de Santiago de Compostela, Santiago de Compostela, Spain
[b] Utrecht University, Utrecht, The Netherlands

ARTICLE INFO

Keywords:
Generalised additive models
Explainable Artificial Intelligence
Interpretable modelling
Neural additive models
Interpretability
Explainability

ABSTRACT

The interpretability of an intelligent model automatically derived from data is a property that can be acted upon with a set of structural constraints that such a model should adhere to. Often these are in contrast with the task objective and it is not straightforward how to explore the balance between model interpretability and performance. In order to allow an interested user to jointly optimise performance and interpretability, we propose a new formulation of Neural Additive Models (NAM) which can be subject to a number of constraints. Accordingly, our approach produces a new model that is called Constrainable NAM (or just CNAM in short) and it allows the specification of different regularisation terms. CNAM is differentiable and is built in such a way that it can be initialised as a solution of an efficient tree-based GAM solver (e.g., Explainable Boosting Machines). From this local optimum the model can then explore solutions with different interpretability-performance tradeoffs according to different definitions of both interpretability and performance. We empirically benchmark the model on 56 datasets against 12 models and observe that on average the proposed CNAM model ranks on the Pareto front of optimal solutions, i.e., models generated by CNAM exhibit a good balance between interpretability and performance. Moreover, we provide two illustrative examples which are aimed to show step by step how CNAM works well for solving classification tasks, but also how it can yield insights when considering regression tasks.

1. Introduction

Today's society is accumulating more and more data whose use can optimise existing processes and bring wealth and knowledge. This motivates a need for tools that can process these data according to the specific tasks at hand.

Machine Learning (ML) and Artificial Intelligence (AI) recently gained lots of success thanks to greater performance with respect to traditional approaches on many benchmark problems [1,2]. The ML and AI approach consists in specifying the end goal and constraints to be satisfied and the programme automatically adjust itself to satisfy those. Unfortunately these models, while effective, are typically not understandable by humans. This is because their main job is only to maximise predictive generalisation, without taking into account the intelligibility of the model itself.

The lack of interpretability of some AI-based systems is an issue for a number of applications for which quality assurance, trust, legal liability and adherence to ethical principles is a must. Motivated by this, a lot of work has been done in the field of eXplainable Artificial Intelligence (XAI) [3,4], a collective effort to explain the behaviour and predictions of ML models to allow debugging, oversight, auditing, knowledge discovery, and safe democratisation of these new powerful information-processing technologies in many domains [5,6].

In this paper we define *explainability* as the ability of a system to give an explanation to the user, that is a report of (part of) the causal reasoning that lead to a particular outcome. We refer to *interpretability* instead of the less demanding property of a system of being inspectable in its parts in a meaningful way. In this sense, interpretability is a prerequisite for explainability.

XAI has evolved in different directions based on the requirements stated by the stakeholders of the explainable systems. For example, in order to build tools that are as widely applicable as possible some researchers have prioritised the development of "black-box explainers" [7]. In this context, no assumption is made on the model and we are only left with the input–output relationship. In that case we have to rely on post-hoc approaches. These typically involve approximating the original model in an appropriate neighbourhood of data with an interpretable model (a so-called white-box) and then inspecting that to provide explanations, as suggested by [8,9] (this approach is also

* Corresponding author.
E-mail address: ettore.mariotti@usc.es (E. Mariotti).

https://doi.org/10.1016/j.inffus.2023.101882
Received 17 October 2022; Received in revised form 2 May 2023; Accepted 6 June 2023
Available online 12 June 2023

known as surrogating). The surrogation paradigm is a very powerful one as it allows in principle to explain any black-box model, even deep neural networks, as long as the surrogating models are powerful enough to act like them. In virtue of this, if we want to enhance the explanation of black boxes, we first need good interpretable models by design so that we can later use them for surrogating.

To put it in other terms:

- we are interested in explaining black box models as these are widely used;
- interpretability is a prerequisite for explainability, by the argument stated above;
- we should aim for models which are interpretable because these make ideal surrogate models for explaining other models.

This opens up the question of what it really means for a model to be interpretable. While with an imperative-style approach (where the programme is hard-coded by a human) the interpretability is kept in each sub-component, with the ML-style approach (as it is usually employed) the only solutions explored are those that solve the task regardless of the final form of the model. A way of leveraging the power of the ML approach is to guide the search of models not only based on their performance but also on some other targets that reflect their interpretability. In this sense we can imagine that an appropriate regularisation term into the target loss could constrain the optimisation of both dimensions (i.e., interpretability and performance).

Just as the appropriate measure for predictive performance varies from task to task, the appropriate constraints for interepretability are also application-dependent (usually penalise complexity and favour sparse representations) and should iteratively be refined with domain experts [10]. Generalised Additive Models (GAM) and Neural Additive Models (NAM) are well-studied classes of models that have powerful predictive performance while retaining an interpretable structure. This work addresses a new formulation for fitting GAM, named Constrainable Neural Additive Model (CNAM), such that appropriate interpretable constraints can be enforced to jointly maximise performance and interpretability. In order to benchmark this model against other approaches we use a novel interpretability metric that is called SHAP-Length [11], which exploits the well-known SHapley Additive exPlanations (SHAP) first introduced by [9].

The main contributions in this work are as follows:

- a novel formulation of NAM, CNAM, such that interpretablilty constraints can be enforced and jointly optimised alongside task-related performance metrics;
- an extensive benchmark of CNAM on 56 binary classification datasets against 13 different models;
- an illustrative use-case of CNAM on a classification dataset;
- an illustrative use-case of CNAM on a regression dataset.

The rest of the manuscript is organised as follows. In Section 2, we discuss related work in the field. In Section 3, we describe the structure and properties of CNAM. In Section 4, we experiment first with a benchmark study for explicitly exploring CNAM interpretability-performance tradeoff on binary classification tasks and then go deeper with two illustrative use cases: an astrophysics classification task and a socio-economic regression task. Finally in Section 6, we draw conclusions and delineate future work.

2. Related work

The history of learning from data can be traced back to early work in the 19th century where, motivated by the desire of predicting astronomical data and minimising reconstruction errors, theoretical foundations and closed-form solutions for Linear Models (LM) were developed, which minimised the Mean Squared Error (MSE) as proposed by [12,13]. The target y is modelled as $\hat{y}$ from a set of features x_i and the task is to find a set of real coefficients β_i such that

$$\hat{y} = \sum_i \beta_i x_i$$

and the MSE= $1/N \sum_j^N (\hat{y_j} - y_j)^2$, is minimised.

Much later, [14] unified some scattered views of modelling a certain target with a LM and introduced Generalised Linear Models (GLM), where the target y is transformed with the so-called *link function* $g(\cdot)$ with static coefficients α_i such that

$$g(y) = \sum_i \alpha_i x_i$$

This new modelling allows (among other things) to have linear *classifiers* by choosing the logistic sigmoid as the link function, leading to what is known as Logistic Regression (LR).

A further generalisation of LM was developed at Bell Labs by [15,16] where the condition of having a static coefficient α_i is relaxed and the relationship is allowed to be a non-linear function $f_i(\cdot)$ of the univariate feature such that

$$g(y) = \sum_i f_i(x_i)$$

This new formulation is appealing because it is more expressive than GLM while keeping a relatively simple and understandable structure. When the model is fitted one can indeed easily visualise each f_i (also called a *shape function*) as a function of the x_i values, naturally providing both a global view of the behaviour of the model on a dataset and a local explanation for the prediction of a single data point.

From a practical point of view, initially the f_i were based on Regression Splines of degree d of the form $f_i(x_i) := \sum_{k=1}^{d} \beta_k b_j(x_i)$ (Spline-GAM) or other Kernel Expansions of the feature x_i. On the one hand, this allowed the injection of expert knowledge when designing the model. On the other hand, the fitting procedure was slow and sometimes did not converge properly. These fitting methods were then shown to be outperformed by [17] with a procedure of bagging and boosting binary decision trees leading to a class of models called Explainable Boosting Machines (EBM). For optimisation of speed and memory EBMs compress the representation of the function f_i with a lookup table, a data structure that bins the feature values and maps each bin to the height of the shape function.

EBMs were then further generalised to allow also pairwise interactions (EB2M) of the form $f(x_i, x_j)$ by [18] and proved to be of value with an application in healthcare [19].

A parallel development has attempted to model the shape functions with neural networks. [20] pioneered the work with the so-called Generalised Additive Neural Networks (GANN), where each f_i was represented as a small neural network. The optimisation process followed an iterative approach and did not make use of backpropagation. GANN were successfully used for example with the aim of improving the performance of credit scoring applications [21]. Recently [22] proposed Neural Additive Models (NAM), a modern reinterpretation of GANN with more neurons, a new activation function ExU and a sophisticated training procedure that included dropout, weight decay, output penalty and feature dropout. They reported competitive task performance compared to EBMs and other models on four datasets. It is worth noting that NAM can be a building block for other models, for example as the backbone of autoencoders. The main drawback of NAM is the need for a careful setup of many hyperparameters and the long training time that is required for the convergence of these models. Moreover with their formulation it is more difficult to introduce expert knowledge in the system and to explicitly require a part of a shape function to have a specific form.

In an effort to put the human in the loop, [23] built GAM-Changer, a framework where fitted GAM can be inspected and changed when deemed appropriate. This enables a domain expert to interact with the system and give some tools on how to correct the model where few

Table 1
Short summary of related models in the literature. CNAM is the new model proposed in this work.

	LM	LR	GLM	Spline-GAM	EBM	EB2M	NAM	CNAM*
Classification tasks		x	x	x	x	x	x	x
Regression tasks	x		x	x	x	x	x	x
Can be backbone of Autoencoders	x						x	x
Can use prior knowledge on target			x	x	x	x	x	x
Can use prior knowledge on data				x				x
Pairwise interactions						x		
Arbitrary initialisation	x	x	x	x				x
Constrainable	x	x						x
Explicitly balancing interpretability-performance								x

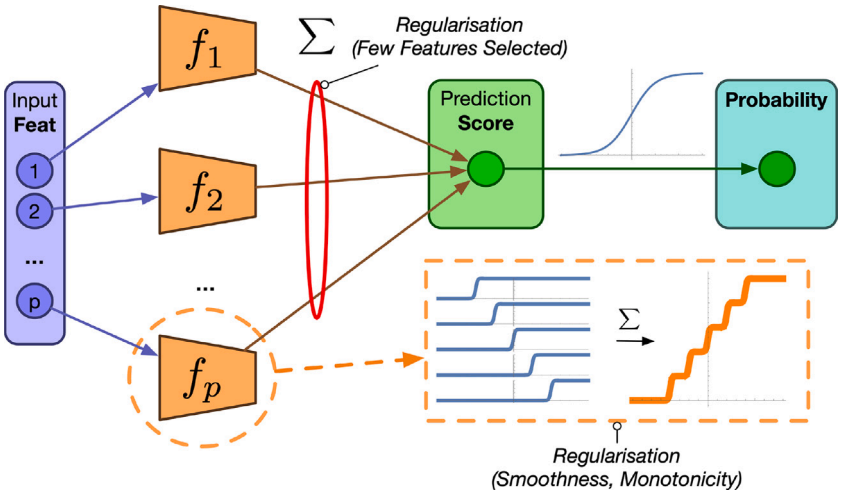


Fig. 1. Schematic view of the structure of CNAM (for classification) and how constraints are inserted. For regression the model is the same with the difference of not having a logistic sigmoid non-linearity at the end.

statistics are available but human knowledge can be of help. While the user does receive feedback on the performance of the modified model, there is currently no way of fine-tuning the changed model keeping fixed the human interventions.

With the aim of overcoming drawbacks of current state of the art models, in this work we introduce the new CNAM, a way of building GAM based on neural networks but with a specific structure such that it is possible to initialise each shape function as an arbitrary curve and to enforce constraints on parts of the network. This is important because it allows us to:

1. leverage the power and speed of EBM which quickly converges to a good solution;
2. leverage the expressiveness of differentiable programming, opening the possibility of jointly optimising performance and interpretability-related metrics.

To conclude this section, Table 1 summarises the main properties of the most outstanding related models in the literature. As we will see in the rest of the paper, CNAM is a promising tool for human-in-the-loop solutions and can in principle be integrated into GAM-Changer enabling a complete feedback-loop between a human and a machine.

3. Model structure

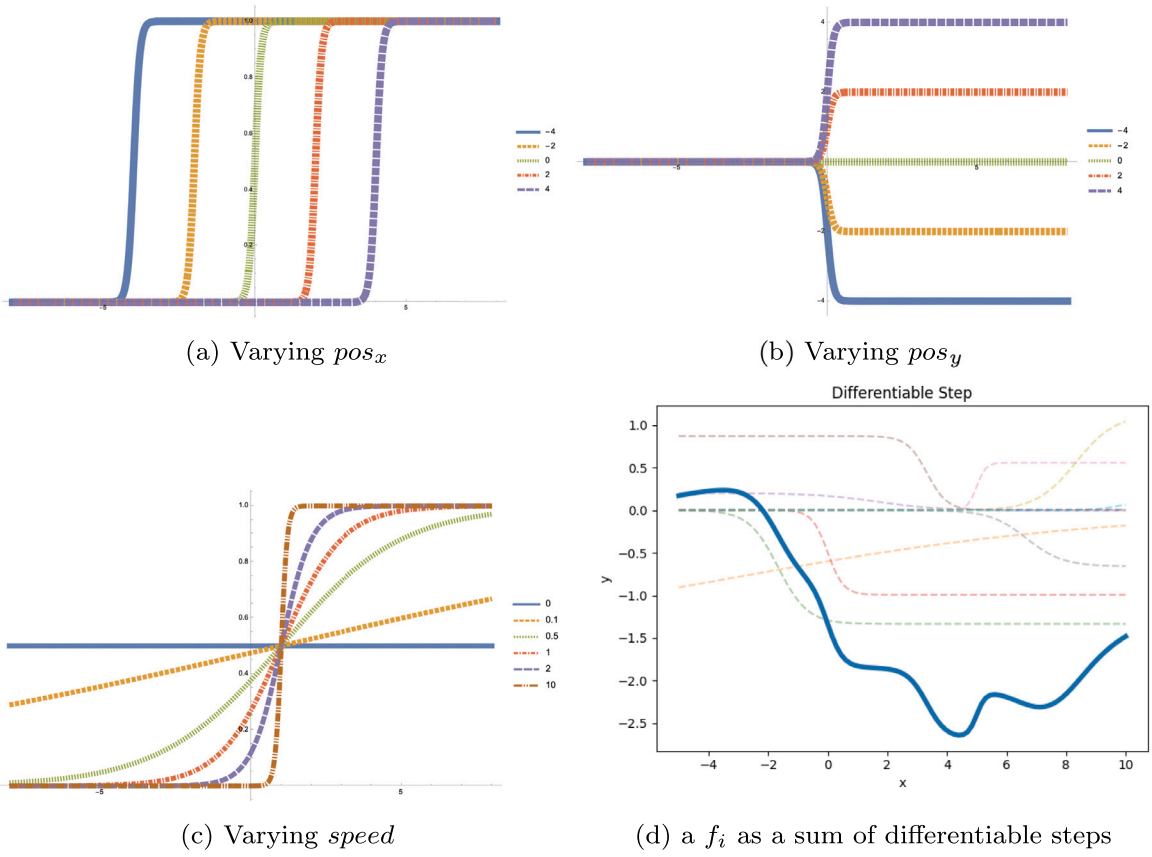
We can graphically represent CNAM as illustrated in Fig. 1. The important novelty of CNAM is the construction of the shape functions f_i as a sum of carefully initialised differentiable versions of the unit step, which we will call *Differentiable Step* (DS). It is worth noting that only a few shape functions are expected to take non-zero values, thus effectively discarding some features from the final score by *Regularisation (Few Features Selected)*. The parametrisation of the DS is what will allow CNAM to be initialised to any arbitrary shape, in particular this property will be used to initialise the model to the solution of EBM (more on this in Section 3.2).

Once initialised, the structure of CNAM is transparent enough that each component has a clear interpretation and this facilitates the formal definition of appropriate regularisers for the problem at hand. As introduced in Section 1, having a model that is flexible enough to enforce different constraints (e.g., sparsity) is key for targeting specific interpretability needs of different use cases. An enumeration of different desirable constraints will be the objective of the following section.

3.1. Desirable constraints

We now proceed to list desirable constraints that could be useful in different situations. These can be appropriate when prior knowledge exists and can help the model to generalise better where data is scarce:

- **Monotonicity** of the shape functions (e.g., in a banking context we might want to enforce that the bigger the "Loan amount term" the lower should be the probability of granting the loan). More formally: for any x_i, x_j such that $x_i \leq x_j$ we have that $f(x_i) \leq f(x_j)$ (positive monotonicity) or $f(x_i) \geq f(x_j)$ (negative monotonicity).
- **Smoothness** of the shape functions. This could be useful as a smooth function is simpler to describe compared to a jagged one (jagged representations are also more likely to be influenced by noise in the data). A user might be interested to explore if a smooth relationship between the feature and its target reasonably model the task at hand. Moreover, since a smooth curve is easier to describe in words than a jagged one, such constraint becomes then useful when the objective is to use natural language in order to provide explanations [24,25].



(a) Varying pos_x (b) Varying pos_y

(c) Varying *speed* (d) a f_i as a sum of differentiable steps

Fig. 2. The effect of changing the parameters of the differentiable steps and how they can form a f_i.

- **Few** shape functions (useful for problems where the number of features is so large that would be impractical to manually assess every shape function). In other words this entails having a sparse representation following the principle of Ockam's razor: "if two models describe the same event, the model with less assumptions (i.e., the model that relies on fewer variables) is preferable". It is possible to mathematically enforce some sparseness of a linear combination of terms x parametrised by the dot product with w resulting in $w.x$ by minimising the L_1 Norm of w, that is: $min \sum_i |w_i|$
- **Local freezing** some part of the shape function (e.g., to help optimise fairness measures). For example if we had a categorical variable describing some protected attribute one might want to enforce that the score for the protected attribute match some formal description of fairness.

It is important to note that the desirable constraints listed above are not an exhaustive list and are context-dependent. Furthermore, some of these constraints may conflict with maximising task performance or even among themselves. For instance, while enforcing monotonicity of a shape function may improve the model's interpretability, it may negatively impact its task performance. Similarly, enforcing sparsity of shape functions may help with interpretability, but at the cost of some loss in predictive accuracy.

Another approach that could be considered is to measure the agreement of feature importances as indicated by a human expert with what the model currently predicts. While this could help the model quickly converge to a well-generalising solution, it should be kept in mind that human ranking could introduce biases that are not present in the data. Developing a suitable metric for measuring such agreement is a topic for future research.

Despite these considerations, the main motivation behind this novel approach is that once the desirable constraints are defined, the optimisation procedure can explore different solutions that aim to satisfy all the given constraints. In the following section, we will discuss how to precisely define each component of CNAM and how to define appropriate regularisers that enforce the desirable constraints.

3.2. Model construction

CNAM specifies the structure of how each shape function can be constructed in a rather straightforward way. Intuitively, we develop the idea that each shape f_i is a sum of j *step functions*, an approximation of indicator functions

$$f_i(x_i) = \sum_j step_j(x_i)$$

This way the model can be rewritten as

$$g(y) = \sum_i \sum_j step_j(x_i)$$

As introduced previously, we want $step(\cdot)$ (the DS) to be a soft version of an indicator function that adds or subtracts a specific (learnable) quantity at a certain position in the shape function. We parametrise DS with three parameters: pos_x, pos_y, $speed$. DS can then be defined as follows:

$$step(x_i) := pos_y * \sigma(speed * (x_i - pos_x))$$

where

- $\sigma(\cdot)$ is the Logistic Sigmoid function, that is: $\sigma(x) = \frac{1}{1+e^{-x}}$;
- pos_x is the position on x where the shape is centred (see Fig. 2(a));
- pos_y is the maximum height of the step (see Fig. 2(b));
- $speed$ is a parameter that specifies how quickly the step grows (see Fig. 2(c)).

When many carefully initialised DS are added up they can form a curve of arbitrary shape (see Fig. 2(d)).

Then, without loss of generality, we can aggregate all the shape functions f_i with a linear layer parametrised by a vector α to produce the final prediction (log-odds in case of classification), as follows:

$$g(y) = \sum_i \alpha_i \cdot f_i(x_i)$$

The previous formulation can be useful for enforcing some regularisation like the L_1 norm on α_i in order to favour a sparse representation (i.e., when few shape functions are desired).

This formulation is well-suited for binary classification tasks and regression tasks. However, we acknowledge that extending our model to other tasks requires additional considerations. One possible approach for non-binary classification is to use a one-vs-rest fashion, where a separate model is trained for each class. This approach increases the complexity of the system, but it can be effective for multiclass classification problems.

3.3. Enforceable constraints

With this formulation we can now begin to impose some of the constraints discussed in Section 3.1 by specifying additional loss terms to be jointly optimised or by imposing some structural changes. Here's how the list of possible enforceable constraints enumerated in Section 3.1 can be implemented:

- **Monotonicity** of the shape functions: This can be achieved by imposing pos_y and $speed$ to be always greater (or smaller) than zero. This can be done by applying the function $max(x, 0)$, also known as ReLU [26,27], to them.
- **Smoothness** of the shape functions: This can be achieved by incentivising small values for the $speed$ parameters of the various DS of a given shape function. This can be done by minimising the L_2 norm of the $speed$, that is to add to the final loss function the component $\lambda\sqrt{\sum_i speed_i^2}$. The parameter λ controls the amount of regularisation. Notice that with this formulation we can have smooth functions that are still able to model big jumps if necessary, something that spline-based GAM failed to achieve because the splines implicitly encode a prior of global smoothness. On the other hand, enforcing an L_2 on the $speed$ still allows local speeds to be very high if appropriate.
- **Few** shape functions: this can be enforced by imposing sparsity (i.e., L_1 regularisation) on the final linear layer that aggregates the shape functions f_i. That is, we can represent without loss of generality the models as $g(x) = \alpha_1 * f_1(x_1) + \cdots + \alpha_p * f_p(x_p)$ and then incentivise a sparse representation by adding to the final loss the L_1 norm of α: $\lambda_2 \sum_i^p |\alpha_i|$, where λ_2 controls the amount of regularisation. This is similar to what [28] proposed with Lasso, but it is now possible to apply it to close-to-optimal EBM solutions, something that up to now was not possible to do.
- **Local freezing**: we can exclude parts of the model from the optimisation process by setting the gradient of a certain step to 0. This prevent any further updates on that parameter and thus any change on that part of the function.

3.4. Parameter initialisation

As already anticipated earlier, one of the key features of CNAM is the ability to be initialised as solutions of faster solvers (e.g. EBM) easily. This is important because if we try to optimise the model from a typical random initialisation (e.g., following the principles delineated by [29]) we observed empirically that the training can be unstable and result in suboptimal solutions with respect to EBM. On the other hand if we optimise the model while starting from the solution found by EBM, then CNAM is able to end up finding solutions that have similar or higher performance while also optimising the interpretability constraints (thus finding a solution that exhibits a better balance between performance and interpretability). This will be shown later with the experiments in Section 4.

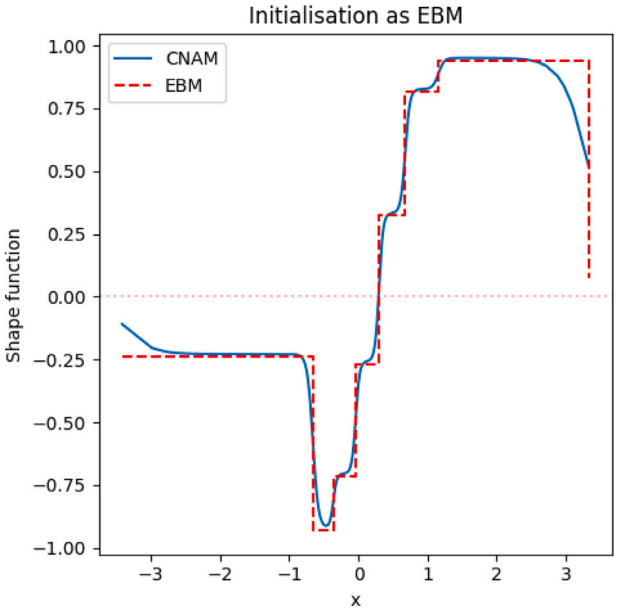


Fig. 3. Example of applying algorithm 1 to some random dummy data with the aim of initialising CNAM as a good approximation of EBM.

For the specific initialisation we set the initial value of $speed$ to a large value (we found 100 to be heurisically a good guess) to better approximate the unit step. The translation parameter pos_x can be set as the x-positions of the lookup table of the shape function of EBM. The height of the shape function (described by pos_y) is finally set to the y-position of the EBM by first initialising all the pos_y to 0 and then adjusting the pos_y from the right-most to the left-most using Algorithm 1. An illustration of how this algorithm works can be seen in Fig. 3.

That said, the random initialisation can still be useful for some use-cases where it is not straightforward to train an EBM (e.g. as components of an Autoencoder). We still suggest to initialise the parameter pos_x for shape i as the quantiles of the univariate distribution of feature i. On the other hand for classification and regression purposes we consistently found in our preliminary experiments that EBM already converges to a close-to-optimal solution and initialising CNAM to that vastly improves the final model across both performance and intelligibility metrics.

Algorithm 1 Initialise CNAM as EBM

Require: y_{EBM}
  $y_{CNAM} \leftarrow y_{EBM} * 0$ ▷ Initialise all as zero
  $y_{CNAM}[-1] \leftarrow y_{EBM}[-1]$
  for $k = 2, k \leq Length(y_{EBM}) + 1, k++$ **do**
    $y_{CNAM}[-k] \leftarrow y_{EBM}[-k]$ ▷ Set the height to the desired one
    $y_{CNAM}[-k+1] \leftarrow y_{CNAM}[-k+1] - y_{CNAM}[-k]$ ▷ fix the effect on the following bin
  end for
  return y_{CNAM}

3.5. Optimisation

Once we have initialised the CNAM parameters, we can optimise the final loss, which includes both task-performance and task-interpretability terms, using a gradient-based optimiser such as ADAM

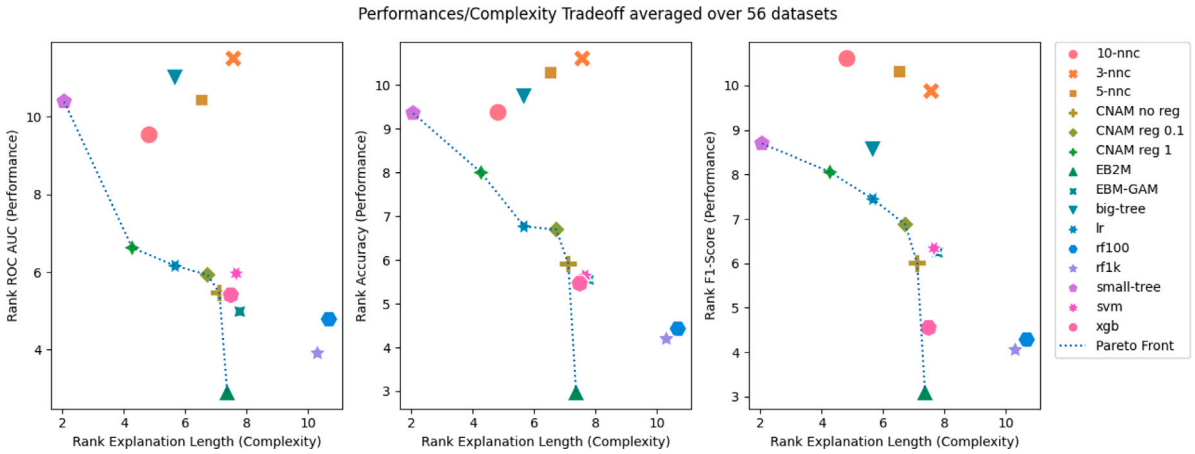


Fig. 4. A benchmark of the performance-interpretability tradeoffs regarding different models on 56 datasets. The x-axis represents the relative **Rank** of interpretability (measured as SHAP-Length) while the y-axis corresponds to the relative **Rank** of performance (measured as ROC AUC, Accuracy, F1-Score). The **best models lie in the lower-left** part of the plots. The dotted blue lines highlight the Pareto front, i.e. the set of non-dominated solutions. Various instantiations of CNAM on average lie on that Pareto front. Interestingly, even without regularisation, CNAM scores better than EBM (without pairwise interactions).

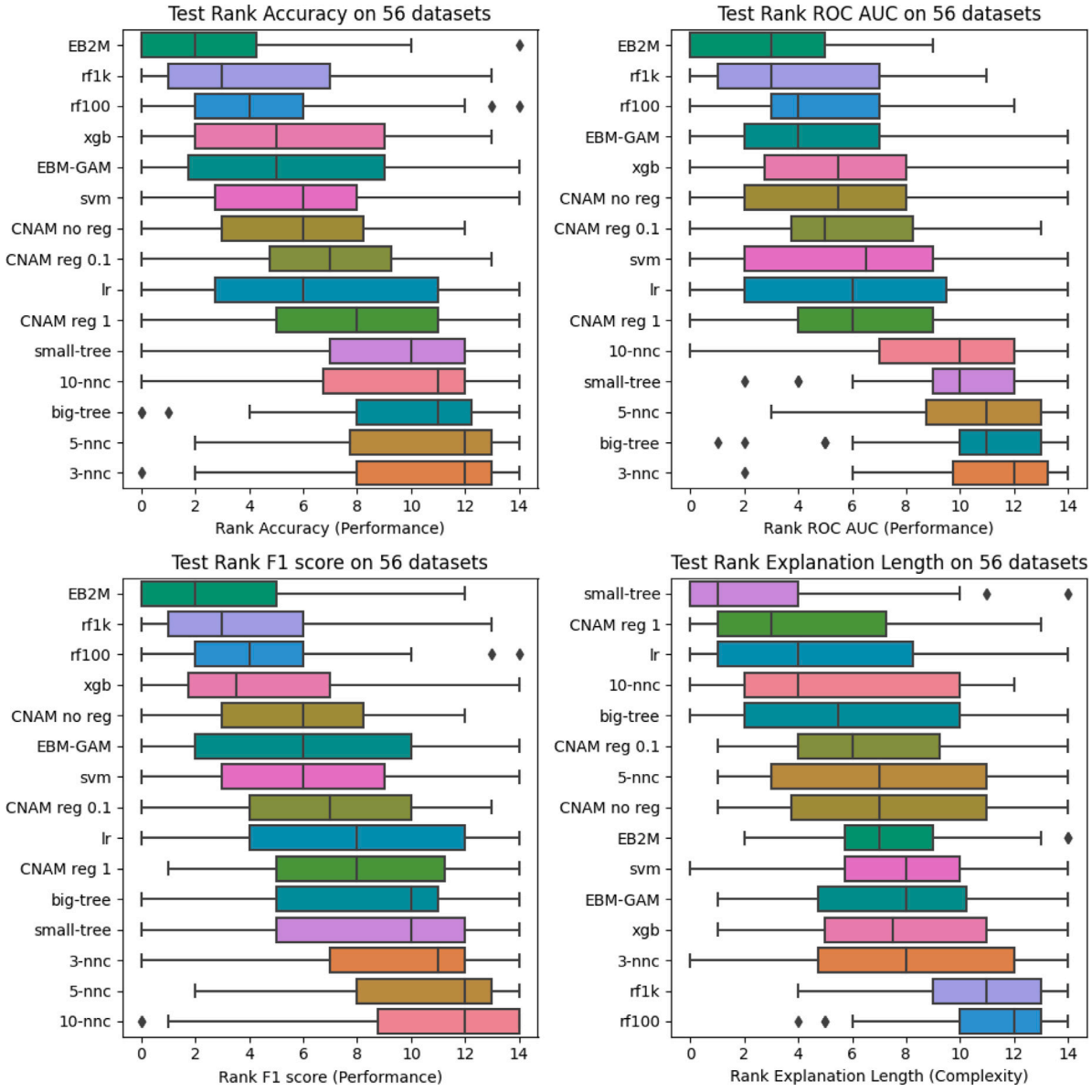


Fig. 5. A box plot of the results of the benchmark for the different models across the different statistics.

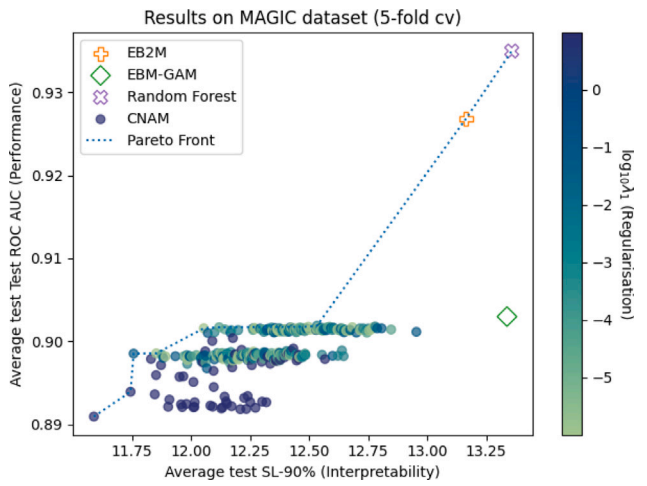


Fig. 6. Performance-Interpretability tradeoff on the MAGIC Telescopes dataset. CNAM configurations are coloured by the amount of λ_1 regularisation, with blue indicating high regularisation and green representing low regularisation. More interpretable models with lower SL (x-axis) and better performance (y-axis) are situated in the upper-left corner of the figure. It is important to note that λ_2 also influences the results, but we have omitted its visualisation for the sake of clarity.

[30]. This technique allows us to iteratively adjust the parameters of CNAM towards a solution that minimises the target loss, even without an explicit closed-form formula for it. To control for overfitting, we periodically evaluate against a validation set and employ an early stopping criterion [31].

4. Experiments

CNAM can be used both for binary classification or regression. We first benchmark the model with different regularisation parameters on 56 binary classification datasets (see Section 4.1). Subsequently, we perform a deep dive into two specific datasets, a classification task (see Section 4.2) and a regression task (see Section 4.3), to provide a more in-depth analysis of the model's behaviour. We implemented CNAM as an open-source software (available online at https://gitlab.nl4xai.eu/ettore.mariotti/cnam), using Python and pytorch [32] as the deep learning backend infrastructure, structuring the code using pytorch-lightning library.

In our experiments, we explore the tradeoff between performance and interpretability by varying the weights of the weighted loss function according to the specific application domain. Although we lack theoretical convergence guarantees, our empirical observations suggest that by trying different coefficients, we can obtain a diverse set of solutions that effectively navigate the performance-interpretability tradeoff. Later in the subsections, we demonstrate this practically with different choices of regularisation parameters as the added term for the loss.

4.1. Binary classification benchmark

In this section we focus our attention to the task of binary classification and we benchmark CNAM against other classifiers across 56 tabular datasets taken from [33]. We measure both task-performance and task-interpretability scores. The datasets are related to different domains and include both real-data as well as simulated-data and spans both in size and class imbalance. The basic properties of each dataset are reported in Table 2.

In order to compare the results we rank the models for each dataset sorting them with respect to the specific metric value they scored. Once this ranking is obtained, we average the rank across all the datasets. We selected three measures of classification performance:

- Accuracy: the fraction (percentage) of correctly classified instances.
- F1-Score: the geometric mean of precision and recall.
- ROC AUC: the area under the receiver operating characteristic curve (i.e., under the ROC curve), that is the curve of true positives versus false positives at all classification thresholds.

For measuring interpretability in terms of model complexity we opted for the so-called SHAP-length metric [11] which is a model-agnostic metric which allows us to compare heterogeneous models. This metric returns the number of SHAP attributions of each data point such that the set of attributions ϕ_i captures a given fraction (we settled at 90% here) of the overall explanation mass $\sum_i |\phi_i|$. More formally: $SL_{90\%} :=$ the smallest i such that $\frac{\sum_i \text{sorted}(|\phi_i|)}{\sum_i |\phi_i|} \leq 0.9$. This intuitively captures the length of the "compressed" explanation of each prediction, that is what a user would have to read in order to get a rough sense of what is the impact of different features. These lengths are then averaged across all the data points of the dataset in order to return a single interpretability score per dataset-model pair. It is important to note that SL might not always be the ideal measure of interpretability. Indeed, the best interpretability metric depends on the context, user preference, and the specific domain of the task. A model with a lower SL will typically have more sparse explanations, so the lower the SL, the less cognitive load burden on the user. This is a key consideration when choosing an interpretability metric, as the goal is to provide users with concise yet meaningful explanations that help them understand the model's behaviour and decision-making process.

As stated before, given the scores of each model on each dataset we can rank each model on the same dataset according to the various metrics. Once we have computed the ranking of all the metrics for all the models across all the datasets we can average them and delineate the Pareto front with the best solutions. The Pareto front is the set of all the Pareto-efficient solutions, that is all those solutions in multi-objective problems where no other solution is better than them in one of the objectives.

We tested CNAM against a pool of 12 different models, most of them implemented by the *scikit-learn* package [34]: Explainable Boosting Machines with pariwise interactions (EB2M) and without interactions (EBM-GAM), rf100 (Random Forest with 100 trees), rf1k (Random Forest with 1000 trees), svm (Support Vector Machine), XGB (eXtremely Gradient Boosting trees), lr (Logistic Regression), small-tree (a decision tree with maximum tree depth of 4), big-tree (a decision tree unconstrained), 3-nnc, 5-nnc, 10-nnc (respectively 3-, 5-, and 10-nearest neighbours classifiers).

In order to evaluate the capabilities of CNAM we evaluated 3 different variations of the proposed model, each with different regularisation coefficients. The regularisation was set up such that a sparsity constraint (L_1 norm on the α) is added to the classical cross entropy loss: $loss = crossentropy + \lambda_1 * \sum_i |\alpha_i|$. This way λ_1 becomes a macroscopic hyperparameter that favours model simplicity at the potential cost of task performance. We benchmarked CNAM with $\lambda_1 = 0$, $\lambda_1 = 0.1$, $\lambda_1 = 1$ in order to show how CNAM behaves with no regularisaion, small regularisation and strong regularisation.

Every model is evaluated with 5-fold stratified cross validation. For each single dataset the models are ordered from the best to the worst thus producing a ranking (lower values thus correspond to better models). The ranking is then averaged across all the datasets and the results are reported in Figs. 4 and 5.

Fig. 4 reports for the sake of clarity and readability only the mean of the computed rankings. In Fig. 5 the interested reader can see further details in the form of box plots that help understand what is the distribution of the rankings.

The results suggest that CNAM unconstrained is equivalent to (if sometimes marginally better than) EBM-GAM. This makes sense as by design CNAM is initialised as a good approximation of EBM-GAM.

Table 2
Basic properties of each dataset tested in our benchmark study.

Dataset	# Instances	# Features	Class imbalance
analcatdata aids	50	4	0.000000
analcatdata asbestos	83	3	0.011758
analcatdata bankruptcy	50	6	0.000000
analcatdata boxing1	120	3	0.090000
analcatdata boxing2	132	3	0.005739
analcatdata creditscore	100	6	0.211600
analcatdata cyyoung8092	97	10	0.255181
analcatdata cyyoung9302	92	10	0.344518
analcatdata fraud	42	11	0.145125
analcatdata japansolvent	52	9	0.001479
analcatdata lawsuit	264	4	0.732840
appendicitis	106	7	0.364543
australian	690	14	0.012132
backache	180	32	0.521605
biomed	209	8	0.079691
breast	699	10	0.096375
breast cancer	286	9	0.164507
breast cancer wisconsin	569	30	0.064940
breast w	699	9	0.096375
buggyCrx	690	15	0.012132
bupa	345	5	0.000412
clean1	476	168	0.016966
cleve	303	13	0.007940
colic	368	22	0.068053
corral	160	6	0.015625
credit a	690	15	0.012132
crx	690	15	0.012132
german	1000	20	0.160000
glass2	163	9	0.004554
heart c	303	13	0.007940
heart h	294	13	0.077792
heart statlog	270	13	0.012346
house votes 84	435	16	0.051795
hungarian	294	13	0.077792
irish	500	5	0.012544
labour	57	16	0.088950
lupus	87	3	0.038182
molecular biology promoters	106	57	0.000000
monk1	556	6	0.000000
monk2	601	6	0.098895
monk3	554	6	0.001577
mux6	128	6	0.000000
parity5	32	5	0.000000
pima	768	8	0.091254
prnn crabs	200	7	0.000000
prnn synth	250	2	0.000000
profb	672	9	0.111111
saheart	462	9	0.094470
sonar	208	60	0.004530
spect	267	22	0.345762
spectf	349	44	0.207560
threeOf9	512	9	0.004944
tic tac toe	958	9	0.094181
vote	435	16	0.051795
wdbc	569	30	0.064940
xd6	973	9	0.114332

Instead, when a regularisation of the form discussed above is applied, we can see how the different models actually loose classification performance but gain in interpretability, returning models on the Pareto front of the best tradeoffs. This is important because it is an indication that the model cannot only retain competitive task performance but can successfully explore the tradeoff between competing constraints in a way that the result lie on the set of Pareto-efficient solutions. Moreover, CNAM reg 1 turns up as the second best from the point of view of interpretability (see the plot related to "Explanation Length" on the bottom right side of Fig. 5), only behind small-tree while it is much better from the performance viewpoint (i.e., CNAM reg 1 is always better ranked than small-tree in the other pictures in Fig. 5).

Finally, it is worth noting that in a specific application a user should experiment with different regularisation parameters (and also specific regularisations fit to the specific domain) according to his or her needs.

4.2. Illustrative example on how to address a classification task: MAGIC dataset

In order to gain more intuition of the behaviour of CNAM on a specific dataset we provide as an application the use of CNAM for the MAGIC Telescopes classification dataset [35], a real-world dataset (taken from the UCI repository [36]) that is much larger than all of those included in the set of the previous benchmark study. In this dataset the task is to distinguish between two different kind of particles (gamma-like vs hadrons) given a representation of the image captured by the telescope. From a physical point of view, a particle enters the atmosphere and produce an elliptic flash of light that is captured by the camera of the telescope. The image is then described by the following 10 numerical features, some of which describe an ellipsoidal shape fitted on the pixels:

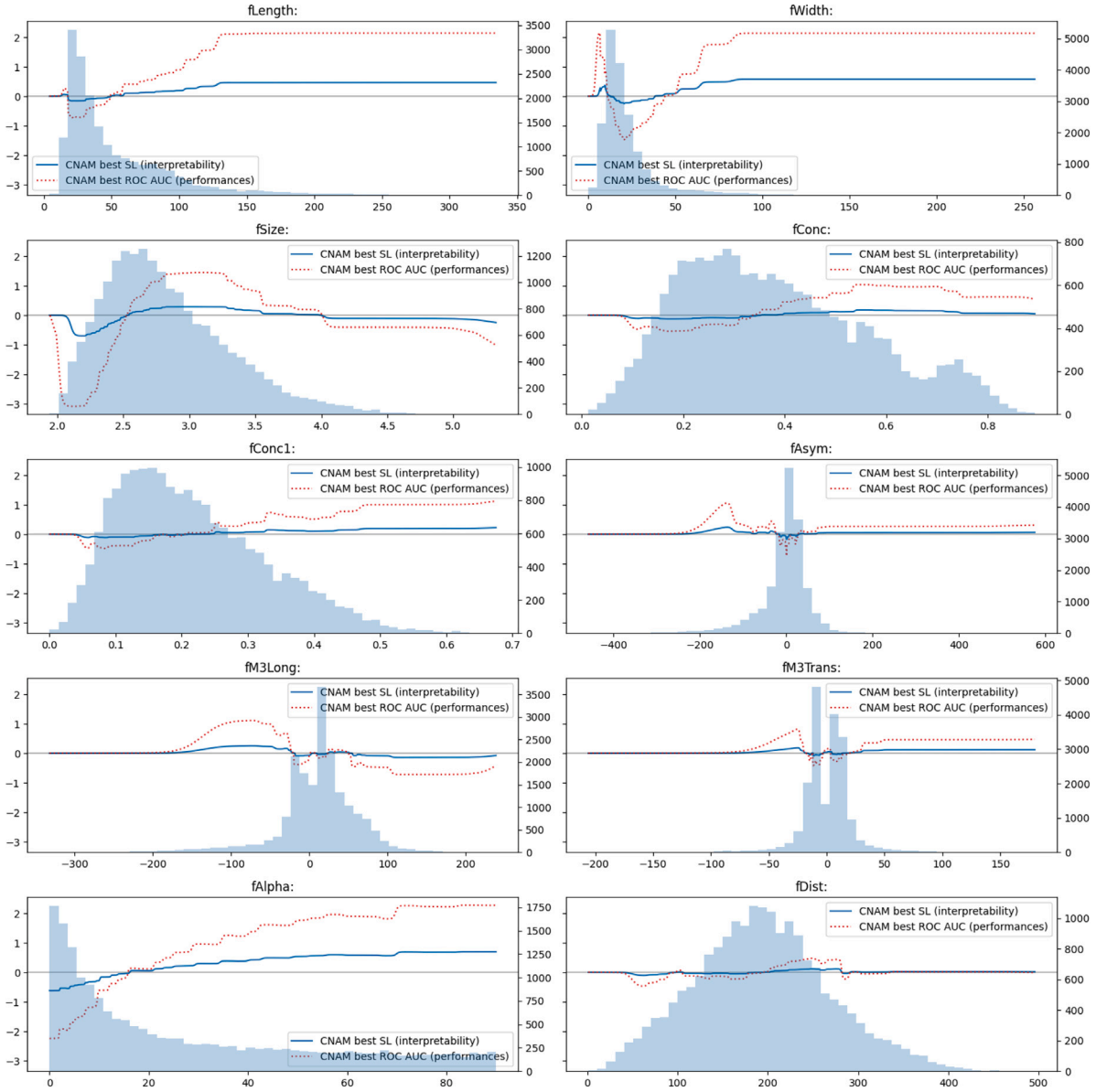


Fig. 7. Visual representation of each shape function of CNAM on the MAGIC Telescopes dataset. Each shape function demonstrates the contribution of a given feature value to the final prediction, which is the sum of the partial scores. The blue solid line represents the most interpretable CNAM instance (**CNAM best SL**), identified as having the shortest Shap Length (SL), while the dotted red line indicates the best-performing instance (**CNAM best ROC AUC**), which has the highest AUC (Area Under The Curve) ROC (Receiver Operating Characteristics) curve score among models trained with varying lambda1 and lambda2 regularisation parameters. The data distribution is visualised as a histogram in the background, with the counts reported on the right-axis of the plot.

- fLength: major axis of ellipse
- fWidth: minor axis of ellipse
- fSize: $\log 10$ of the sum of content of all pixels (photon count)
- fConc: the ratio of sum of two highest pixels over fSize
- fConc1: the ratio of highest pixel over fSize
- fAsym: distance from highest pixel to centre, projected onto major axis
- fM3Long: 3rd root of third moment along major axis
- fM3Trans: 3rd root of third moment along minor axis
- fAlpha: angle of major axis with vector to origin
- fDist: distance from origin to centre of ellipse

The dataset has 19020 instances (12332 gamma and 6688 hadron), with a class imbalance of 0.542. In this task the Accuracy metric is not meaningful as classifying a hadron (background) as a gamma (signal) is worse than vice versa (classifying the signal as background). For comparing different classifier a more sensible metric is ROC AUC, as it allows the evaluation of the performances at different classifying thresholds.

For illustrative purposes we will fit CNAM with different regularisation parameters, namely λ_1 (encouraging sparsity, that is the shape functions are more often to 0) and λ_2 (encouraging smooth shapes), drawing them randomly from a log-uniform distribution $Y = \log(X)$ where $X = U(-6, 1)$. Each candidate configuration is evaluated with a 5-fold stratified cross-validation with the ROC AUC score (performance) and $SL_{90\%}$ (interpretability). For comparison we will also fit other competitive models: EBM-GAM (as it is CNAM initialisation), EB2M and Random Forest with 1000 trees (as both models demonstrated high performances on the benchmark study). In Fig. 6 is shown how each model scores in the performance-interpretability space.

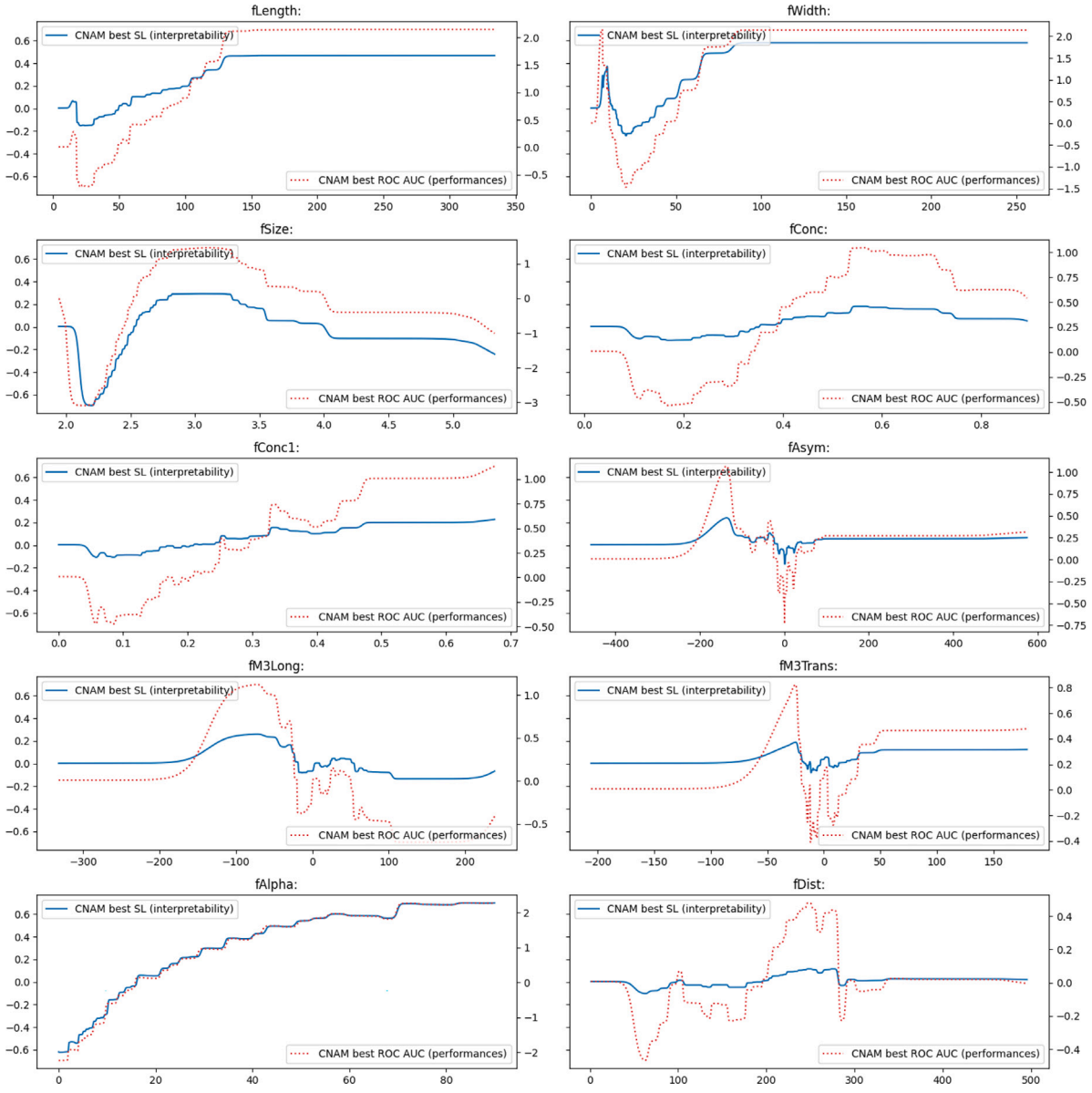


Fig. 8. A modified version of Fig. 7, with each shape function adjusted for scale and without the histograms in the background for ease of comparison between the CNAM best SL and CNAM best ROC AUC models. In each plot, the left axis represents the log-odds contribution of the CNAM best SL (blue solid line), while the right axis corresponds to the contribution of the CNAM best ROC AUC (red dotted line).

It is interesting to notice how EB2M and Random Forest both perform significantly better in terms of average ROC AUC than EBM-GAM and CNAM, suggesting that higher-order interactions between features are relevant for the specific task. That said, CNAM solutions have very similar performances as EBM-GAM, while being more interpretable (having a smaller $SL_{90\%}$).

To understand how the model behaves globally (global explanation) it is possible to visualise each univariate shape function of CNAM as a function of the feature values. Indeed in Fig. 7 it is possible to see such an explanation. The blue histogram describes the empirical distribution of data (with counts mapped on the right axis of each plot) while the two lines in each ax show the shape functions of the most interpretable (**CNAM best SL**) and of the best-performing CNAM solution (**CNAM best ROC AUC**) of the Pareto front which is depicted in Fig. 6.

As it is, it is not straightforward how to make a comparison of the two different solutions as they might have a different fit intercept on the final linear layer. For better appreciating the difference between the solutions, Fig. 8 shows them scaled (the contribution of **CNAM best SL** is marked on the left axis, the one of **CNAM best ROC AUC** is marked on the right axis). Both models rely heavily on the *fAlpha* feature in a remarkably similar way, but while **CNAM best ROC AUC** also take into account other quantities, **CNAM best SL** sacrifices the use of some features (like *fDist*, *fConc*, *fConc1*, *fAsym*) in order to gain more interpretability while keeping performance reasonable. This could be useful information for the data scientist working on the task, as he or she might decide to refine the measurement of *fAlpha* or introduce new features that describe a similar physical phenomenon. It could be argued that by reducing the reliance on some features it is possible to obtain a more interpretable model at the price of some performance, but this can still be useful for directing the attention of the user to relevant information that can enhance globally the pipeline of the experiment (that is, measuring new things, discover biases or problems in simulations and in general design better experiments).

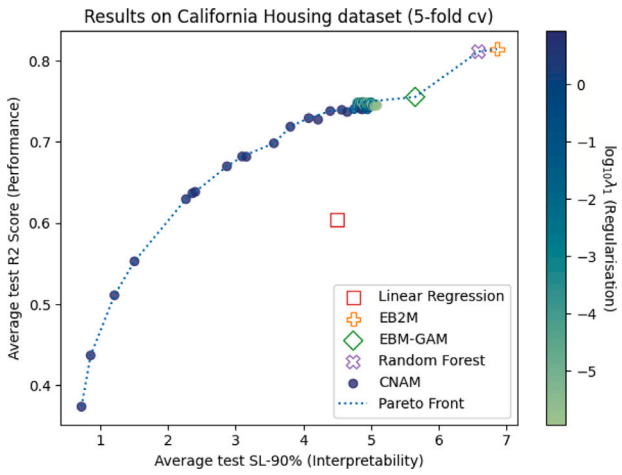


Fig. 9. Performance-Interpretability tradeoff on the California Housing dataset. CNAM configurations are coloured by the about of regularisation, blue means high regularisation while green means low regularisation. As a reminder, the best models would be in the upper-left corner of the figure.

Still, as remarked earlier, the significantly higher performance of Random Forest and EB2M suggests that restricting the modelling to only the univariate components (as EBM-GAM and CNAM do) might be inappropriate for this specific task. The final choice of which class of models to use depends on the data and the impact that such an intelligent system has on the world. While a physicist might be comfortable with not knowing why a given event was labelled as either gamma-like or hadron (thus opting for a black box with higher performances), a doctor might trust more on a less performing yet more interpretable model for a diagnostic system that should guide his or her decision process.

4.3. Illustrative example on how to address a regression task: California housing dataset

In order to show how CNAM can be used not only for solving and gaining insights in classification problems but also for regression tasks, in this section we go in depth with the California Housing dataset [37]. This dataset, derived from the 1990 U.S. census, describes each California district based on 9 numerical features and aims at predicting the median house value of that district. It has 20640 instances. The features are the following:

- MedInc: the median income in a district.
- HouseAge: the median age of the house in a district.
- AveRooms: the average number of rooms per household.
- Population: the population of the district.
- AveOccup: the average number of household members.
- Latitude: the latitude of the district.
- Longitude: the longitude of the district.

The target to be modelled is the median house value for each California district (in 100k $).

We fit the model following the setup described earlier in the classification experiment. One significant difference is the performance metric, which in this context is the R^2 score. The R^2 score, also known as *coefficient of determination*, is a regression metric that represents the target proportion of variance that has been explained by the features and is defined as $R^2(y, \hat{y}) = 1 - \frac{\sum_i y_i - \hat{y}_i}{\sum_i y_i - \frac{\sum_i^n y_i}{n}}$. In linear settings, it is equivalent to the square of a correlation coefficient. Our focus is on finding different solutions that explore the tradeoff between the two quality metrics (i.e., R^2 and $SL_{90\%}$). We also fit an LR model (for its simplicity), EBM-GAM, EB2M and Random Forest that will serve as baselines for comparison, as we did in the previous illustrative example (Section 4.2).

Fig. 9 summarises graphically the result of the experiment. Interestingly, CNAM offers a set of solutions belonging to the Pareto front of optimal solutions in terms of $SL_{90\%}$ and R^2 score. Notably, we found solutions that have a similar R^2 score as LR while being almost twice as interpretable. Conversely, we also find solutions that have the same interpretability as LR while performing significantly better (R^2=0.75 vs R^2=0.6, i.e., an improvement of 25%).

In Fig. 10 we plotted the shape functions of the three CNAM configurations chosen from the Pareto front of the earlier experiment, similarly to what was done in Fig. 7. The solid blue line (**CNAM best SL**) is the model that has the same R^2 score of LR while being roughly twice more interpretable in terms of $SL_{90\%}$. The red dotted line (**CNAM best** R^2) is the model that reached the highest R^2 score while the green dash-dotted line (**CNAM Smooth**) is a configuration where smoothness was encouraged thanks to the regularisation of the *speed* parameters. It is possible to see how the shape functions of **CNAM best SL** are more often close to zero, de facto ignoring features such as HouseAge, AveBedrms, Population, giving less importance to the geographical location (Latitude and Longitude are less prominent) and instead relying a lot mainly on MedInc. **CNAM best** R^2 is more jagged and uses more features in a precise way, this way it can closely model what the data has to say and can achieve a relatively higher R^2. **CNAM Smooth** has an in-between representation with the characteristic of being smoother. It can be argued that the smoothness of the shape functions gives an overall easier-to-describe global explanation.

5. Limitations

In our experiments, we employed the SHAP-Length metric to assess the interpretability of models with diverse structures. Future developments might yield alternative metrics better suited for specific use cases or expert preferences, allowing for more tailored evaluations.

CNAM's applicability to various data types presents some challenges and considerations. The key requirement for maintaining interpretability is the use of interpretable features. Adapting CNAM to other tasks involving different data types, like images, text, or time series, requires representing these data types in tabular format through feature engineering. A coherent explanation of the model's predictions might be difficult to derive when applying CNAM directly to pixel-level data in images or to neural network embeddings of text or time series data.

In the context of data fusion, the key to preserving interpretability in a CNAM model lies in the effective transformation of unstructured data into an interpretable tabular format. Through representation learning or feature engineering techniques, unstructured data can be converted into a tabular form with interpretable features, allowing for the effective application of CNAM to data fusion tasks.

Finally, CNAM is restricted to univariate feature modelling, which disregards higher-order feature interactions. Although this simplification allows for easy-to-understand, interpretable models, it inevitably constrains the expressiveness of the model when faced with complex interactions between features. Balancing this trade-off remains a challenge and a potential direction for future research. Feature engineering techniques that explicitly encode known interactions or domain-specific knowledge could be incorporated into the tabular data before applying CNAM. Alternatively, research could explore the development of constrained higher-order interaction terms that maintain some level of interpretability, although this would require a careful design for finding out the right balance between expressiveness and understandability.

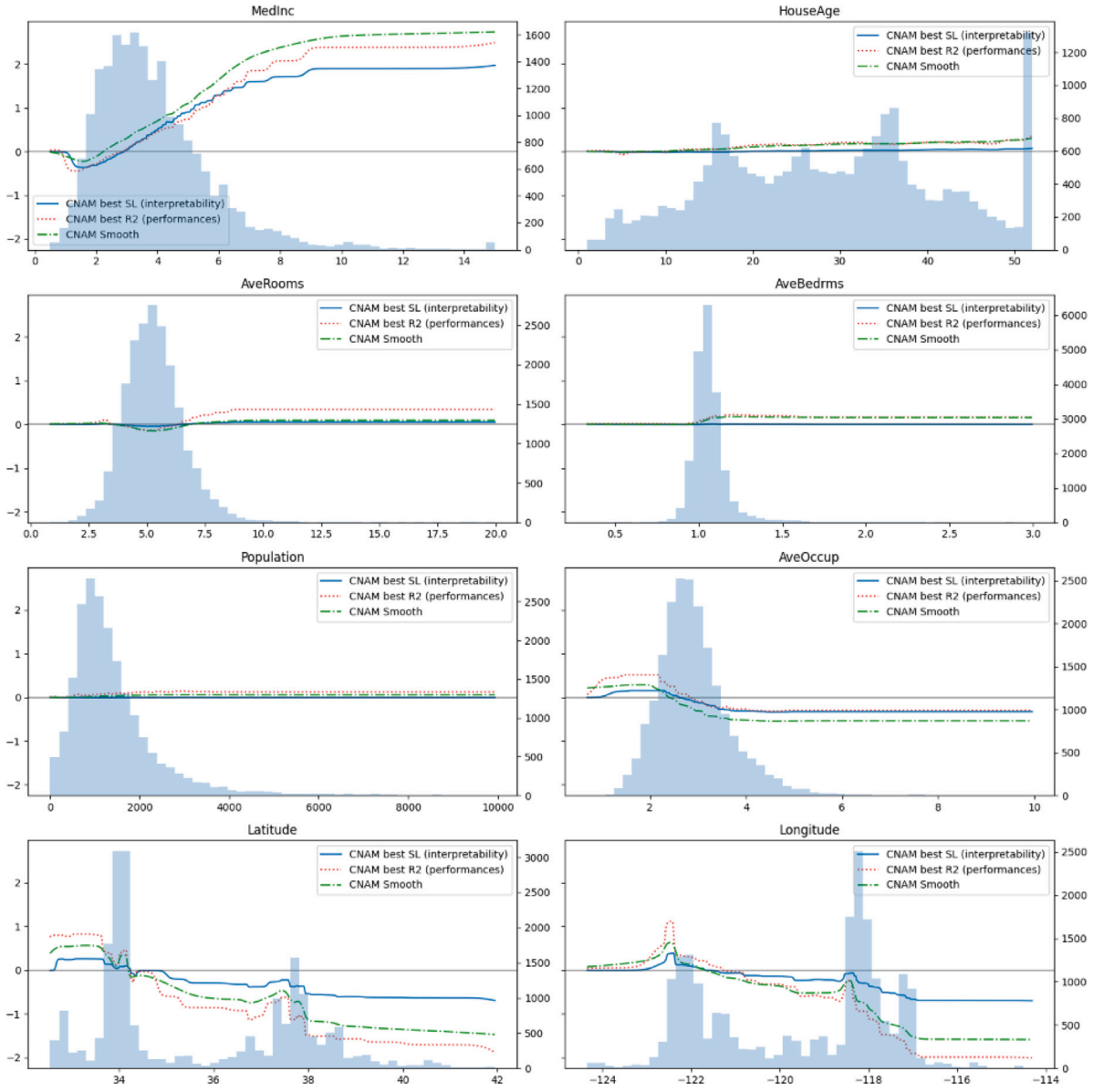


Fig. 10. The shape functions of three different models fitted on the California Housing dataset with different constraints. The dotted red line represents the solution that only maximised the performances on the task (R^2 score). The blue solid one is the model that balanced complexity and interpretability (measured with $SL_{90\%}$). The green dot-dashed line instead is the result of a model with a penalty that encouraged a smooth representation (a useful property for example when verbalising the plot is of interest). Note: for better readability we excluded from the plot data points with extreme values (i.e. districts with vacation resorts with a high number of empty rooms and few resident households).

6. Conclusions and future work

This work introduces CNAM, a constrainable NAM that can be initialised to match an arbitrary function. Its structure allows for inspection and, if needed, constraining to achieve a more interpretable behaviour. By adjusting various regularisation coefficients, it is possible to explicitly explore the performance-interpretability tradeoff. We benchmarked the model on 56 classification datasets against 12 models and observed how CNAM on average finds sweet spots on the Pareto front of multi-objective solutions. We illustrated the effectiveness of CNAM on a specific classification task and a regression task, highlighting the balance between performance and interpretability. It is worth noting that CNAM is available as open source software at https://gitlab.nl4xai.eu/ettore.mariotti/cnam.

As future work, more specific constraints could be designed to accommodate specific needs. For example, it would be of interest to give the model the ability to express an explanation of its behaviour with natural language, rather than just with the visual modality. Another promising direction would be to enhance the human–computer interaction via integration with GAM-changer [23]. This way, both the model and the user could benefit from a feedback loop that iteratively refines the desired objectives in terms of performance and interpretability. In addition, exploring more intelligent search techniques, for instance by resorting to multiobjective metaheuristics, would be a valuable direction for future research to further improve the model's capability in the search for better tradeoffs in the Pareto space.

Efficiently creating counterfactual explanations [38] using CNAM by exploiting the specific structure of the model is another potential research direction. Additionally, it would be intriguing to explore whether a differentiable version of diverse and compelling counterfactuals properties (as proposed in [39]) could be formalised and subsequently incorporated into CNAM.

Finally, enabling pairwise interactions could significantly boost the task performance, albeit at the cost of harder interpretability.

CRediT authorship contribution statement

Ettore Mariotti: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Visualization. **José María Alonso Moral:** Conceptualization, Writing – original draft, Writing – review & editing, Supervision. **Albert Gatt:** Conceptualization, Writing – original draft, Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work is conducted within the NL4XAI project which has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 860621. This work was also supported by the Spanish Ministry of Science, Innovation and Universities (grants PID2021-123152OB-C21, TED2021-130295B-C33 and RED2022-134315-T) and the Galician Ministry of Culture, Education, Professional Training and University (grants ED431G2019/04 and ED431C2022/19). All grants were co-funded by the European Regional Development Fund (ERDF/FEDER program).

References

[1] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, Demis Hassabis, Mastering the game of go with deep neural networks and tree search, Nature 529 (7587) (2016) 484–489.

[2] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A.A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, Demis Hassabis, Highly accurate protein structure prediction with AlphaFold, Nature 596 (7873) (2021) 583–589, Publisher: Nature Publishing Group.

[3] David Gunning, David Aha, DARPA's explainable artificial intelligence (XAI) program, AI Mag. 40 (2) (2019) 44–58.

[4] Edwin Lughofer, Evolving Fuzzy and Neuro-Fuzzy Systems: Fundamentals, Stability, Explainability, Useability, and Applications, Handbook on Computer Learning and Intelligence, World Scientific, 2021, pp. 133–234.

[5] Augusto Anguita-Ruiz, Alberto Segura-Delgado, Rafael Alcalá, Concepción M. Aguilera, Jesús Alcalá-Fdez, eXplainable Artificial Intelligence (XAI) for the identification of biologically relevant gene expression patterns in longitudinal human studies, insights from obesity research, PLoS Comput. Biol. (4) (2020).

[6] Alessandro Renda, Pietro Ducange, Francesco Marcelloni, Dario Sabella, Miltiadis C. Filippou, Giovanni Nardini, Giovanni Stea, Antonio Virdis, Davide Micheli, Damiano Rapone, Leonardo Gomes Baltar, Federated learning of explainable AI models in 6G systems: Towards secure and automated vehicle networking, Information 13 (8) (2022) 395.

[7] Francesco Bodria, Fosca Giannotti, Riccardo Guidotti, Francesca Naretto, Dino Pedreschi, Salvatore Rinzivillo, Benchmarking and Survey of Explanation Methods for Black Box Models, Technical Report, 2021, arXiv:2102.13076 [cs] type: article.

[8] Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin, "Why should I trust you?": Explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 1135–1144.

[9] Scott M. Lundberg, Su-In Lee, A unified approach to interpreting model predictions, in: Advances in Neural Information Processing Systems, vol. 30, Curran Associates, Inc, 2017.

[10] Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, Chudi Zhong, Interpretable Machine Learning: Fundamental Principles and 10 Grand Challenges, 2021, arXiv:2103.11251.

[11] Ettore Mariotti, Jose M. Alonso-Moral, Albert Gatt, Measuring model understandability by means of Shapley additive explanations, in: 2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, Padua, Italy, 2022, pp. 1–8.

[12] Adrien Marie Legendre, Nouvelles méthodes pour la détermination des orbites des comètes, 1805.

[13] Carl Friedrich Gauss, Disquisitiones variae, 1808.

[14] J.A. Nelder, R.W.M. Wedderburn, Generalized linear models, J. Roy. Statist. Soc. Ser. A (General) 135 (3) (1972) 370–384, Publisher: [Royal Statistical Society, Wiley].

[15] Trevor Hastie, Robert Tibshirani, Non-parametric logistic and proportional odds regression, J. Roy. Statist. Soc. Ser. C (Appl. Stat.) 36 (3) (1987) 260–276.

[16] Trevor Hastie, Robert Tibshirani, Generalized additive models for medical research, Stat. Methods Med. Res. 4 (3) (1995) 187–196.

[17] Yin Lou, Rich Caruana, Johannes Gehrke, Intelligible models for classification and regression, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '12, ACM Press, Beijing, China, 2012, p. 150.

[18] Yin Lou, Rich Caruana, Johannes Gehrke, Giles Hooker, Accurate intelligible models with pairwise interactions, in: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '13, Association for Computing Machinery, New York, NY, USA, 2013, pp. 623–631.

[19] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, Noemie Elhadad, Intelligible models for HealthCare: Predicting pneumonia risk and hospital 30-day readmission, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15, Pages 1721–1730, New York, NY, USA, Association for Computing Machinery, 2015.

[20] William J.E. Potts, Generalized additive neural networks, in: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '99, Pages 194–200, New York, NY, USA, Association for Computing Machinery, 1999.

[21] Da De Waal, An Investigation Into the Use of Generalized Additive Neural Networks in Credit Scoring, 2005, pp. 1–10.

[22] Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Ben Lengerich, Rich Caruana, Geoffrey E. Hinton, Neural additive models: Interpretable machine learning with neural nets, in: Advances in Neural Information Processing Systems, vol. 34, Curran Associates, Inc, 2021, pp. 4699–4711.

[23] Zijie J. Wang, Alex Kale, Harsha Nori, Peter Stella, Mark Nunnally, Duen Horng Chau, Mihaela Vorvoreanu, Jennifer Wortman Vaughan, Rich Caruana, GAM Changer: Editing Generalized Additive Models with Interactive Visualization, 2021.

[24] Niland David-Paul, Natural Language Explanations, 2022.

[25] Jose M. Alonso, Senen Barro, Alberto Bugarın, Kees van Deemter, Claire Gardent, Albert Gatt, Carles Sierra, Nava Tintarev, Katarzyna Budzynska, Interactive Natural Language Technology for Explainable Artificial Intelligence, 2019, p. 6.

[26] Kunihiko Fukushima, Cognitron: A self-organizing multilayered neural network, Biol. Cybernet. 20 (3) (1975) 121–136.

[27] Vinod Nair, Geoffrey E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10, Omni Press, Madison, WI, USA, 2010, pp. 807–814.

[28] Robert Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Ser. B Stat. Methodol. 58 (1) (1996) 267–288.

[29] Xavier Glorot, Yoshua Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, in: JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[30] Diederik P. Kingma, Jimmy Ba, Adam: A method for stochastic optimization, in: Yoshua Bengio, Yann LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, in: Conference Track Proceedings, 2015.

[31] Rich Caruana, Steve Lawrence, C. Giles, Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping, in: Advances in Neural Information Processing Systems, vol. 13, MIT Press, 2000.

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, Soumith Chintala, PyTorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems, vol. 32, Curran Associates, Inc, 2019.

[33] Randal S. Olson, William La Cava, Patryk Orzechowski, Ryan J. Urbanowicz, Jason H. Moore, PMLB: a large benchmark suite for machine learning evaluation and comparison, BioData Min. 10 (1) (2017) 36.

[34] Fabian Pedregosa, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Scikit-learn: Machine learning in python, J. Mach. Learn. Res. 12 (85) (2011) 2825–2830.

[35] R.K Bock, A Chilingarian, M Gaug, F Hakl, T Hengstebeck, M Jiřina, J Klaschka, E Kotrč, P Savický, S Towers, A Vaiciulis, W. Wittek, Methods for multidimensional event classification: a case study using images from a Cherenkov gamma-ray telescope, Nucl. Instrum. Methods Phys. Res. A 516 (2–3) (2004) 511–528.

[36] Dua Dheeru, Casey Graff, UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences, 2017.

[37] R Kelley Pace, Ronald Barry, Sparse spatial autoregressions, Statist. Probab. Lett. 33 (3) (1997) 291–297.

[38] Ilia Stepin, Jose M. Alonso, Alejandro Catala, Martín Pereira-Fariña, A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence, IEEE Access 9 (2021) 11974–12001, Conference Name: IEEE Access.

[39] Javier Del Ser, Alejandro Barredo-Arrieta, Natalia Díaz-Rodríguez, Francisco Herrera, Andreas Holzinger, Exploring the Trade-Off Between Plausibility, Change Intensity and Adversarial Power in Counterfactual Explanations using Multi-Objective Optimization, 2022, arXiv:2205.10232 [cs].