Interpretable hierarchical symbolic regression for safety-critical systems with an application to highway crash prediction

Thomas Veran[a,b,*], Pierre-Edouard Portier[a], François Fouquet[b]

[a] *Univ Lyon, INSA Lyon, CNRS, LIRIS UMR5205, 20 Avenue Albert Einstein, Villeurbanne, F-69621, Rhône, France*
[b] *Data New Road, 76 Bd du 11 Novembre 1918, Villeurbanne, 69100, Rhône, France*

Abstract

We introduce a framework to discover interpretable regression models for high-stakes decision making in the context of safety-critical systems. The core of our proposal is a multi-objective hierarchical symbolic regression algorithm able to compute cluster-specific rankings of regression models ordered by increasing complexity. We discover the hierarchical structure by clustering the features' importances of a post-hoc explainability framework (viz., SHAP) applied to a highly flexible predictive model (viz., XGBoost). We rely on a symbolic regression algorithm based on the simulated annealing meta-heuristic to infer sparse linear models which may include non-linear effects (e.g., log-transforms, multiplicative interactions...). This search is guided by two objectives: maximizing predictive performance and minimizing complexity. It ends on a list of Pareto-optimal models that fosters a dynamic interpretative process: the user navigates from the least to the most complex model and decides the ones he can trust depending on whether he understands them, and whether he is satisfied by the quantified uncertainty of their parameters and predictions. Our approach achieves promising results when compared to more than ten other interpretable or black-box predictive models on eleven public regression datasets of various volumes, dimensionalities or domains, and on a proprietary dataset for highway crash prediction. On this last dataset, we demonstrate the usefulness of our new ranking-by-complexity of inherently interpretable models.

Keywords: Machine Learning, model interpretability, symbolic regression, multi-objective optimization

*Corresponding author.
Email address: thomas.veran@insa-lyon.fr (Thomas Veran)

1. Introduction

To make high-stakes decisions in safety critical systems based on the output of a data-driven predictive model, it is necessary to consider if the model is trustworthy. Interpretability helps assessing trustworthiness by making clear what the model knows when it makes a decision. Indeed, a predictive model is interpretable when the process leading to a prediction is understandable to humans (Rudin et al., 2022). Elucidation of trustworthiness is also facilitated by stressing what a predictive model does not know, through a clear quantification of uncertainty (Tomsett et al., 2020).

The constraints that make a predictive model interpretable are domain-specific (Rudin, 2019). In this work, we choose to focus on regression analysis for safety critical systems with access to tabular data where the explanatory variables are meaningful. In that context, we consider that the prediction process is fully understandable when it is expressed as a simple formula of the observed variables.

However, for tabular data, given a sufficiently rich hypothesis space, many models can approach the minimum error rate. This phenomenon is referred to as the Rashomon effect (Breiman, 2001). Also, due to various inductive bias, important associations between observed variables and target can often be indistinguishable from spurious associations specific to the dataset (Teney et al., 2022). Therefore, models with the minimum error rate are not necessarily the best suited to help decision-making. In the absence of precise prior knowledge of the conditional independencies relationships between the variables, it may be better to discover a set of potential predictive models and to let the user decide which of them is more trustworthy.

Based on these observations, we propose a framework that fosters a dynamic interpretative process by computing a small subset of cluster-specific models from a large hypothesis space. Our approach can be broken down into two main steps.

Firstly, we consider that data can often be partitioned so that refined predictive models apply to different parts more efficiently and more meaningfully than a global model. We discover this structure by clustering the instances based on the features' scores returned by the SHAP (Lundberg and Lee, 2017) post-hoc analysis of a flexible non-parametric model.

Secondly, we design a symbolic regression (La Cava et al., 2021) algorithm based on simulated annealing to explore an hypothesis space made of simple mathematical expressions that correspond to expansions of linear models with the possible addition of some transforms of the original variables (e.g. compositions of log-transforms, multiplicative interactions, etc.) and with the use of a regularization term to control the bias/variance trade-off. To explore the hypothesis space, the meta-heuristic search conducts a multi-objective optimization (Stinstra et al., 2008) on both a predictive performance metric and a complexity metric. The complexity metric promotes interpretable models, especially by rewarding sparseness and by penalizing colinearities.

The search ends with a list of Pareto optimal models. These models are learned by bayesian inference which ensures that they are formed through an interpretable generative process and that they offer a clear quantification of uncertainty. The user can then navigate among these models, from the least to the most complex, and decide if he can trust a given model depending on whether he understands it, and whether he is satisfied by the quantified uncertainty of its parameters and predictions.

We test our framework on twelve datasets, covering a broad variety of contexts in terms of volume, data types and domains. Eleven of them are public regression datasets. The last is a proprietary dataset in the domain of highway safety analysis where the task is to build a crash prediction model. We obtain very promising results. Our framework outperforms fully interpretable models such as linear models or shallow decision trees while getting close to non-parametric models. In addition, through a realistic case study conducted on the highway network dataset, we demonstrate how our framework enables a dynamic interpretative process that can help field experts develop new safety policies.

The rest of the paper is organized as follows. Section 2 introduces the related work on crash predictions models, model interpretability and symbolic regression. Section 3 describes the proposed method, from the automatic discovery of a hierarchical structure in the data, to the elaboration of Pareto optimal models for each cluster. Experimental results are discussed in Section 4. Finally, in Section 5, we illustrate on the highway network dataset the main components of the dynamic interpretative process made possible by our hierarchical symbolic regression models.

2. Related Work

In this section, we first introduce key concepts of model interpretability. Then, we present the main approaches used for crash prediction modeling, a representative case of safety-critical systems, with a focus on their interpretability. Finally, we introduce references on symbolic regression, the strategy we used to model crash data.

2.1. Model interpretability

In many sensitive area such as highway safety, AI systems are being used to assist field experts in making high stake decisions which may indirectly affect humans' lives. Therefore, stakeholders expect these systems to compel to several properties such as trustworthiness, confidence, fairness, accessibility and interactivity (Arrieta et al., 2020). Some predictive models, such as rule-based systems, generalized linear models (GLM) (McCullagh and Nelder, 2019), generalized additive models (GAM) (Hastie and Tibshirani, 2017) or shallow decision trees are commonly considered as being inherently interpretable and, indeed, they meet all of the above criteria. Other models, such as ensemble of decision trees or deep neural networks, are able to produce highly

flexible decision boundaries and can therefore reach better predictive performance on some datasets. However, they tend to work like black boxes. Post-hoc explanations methods are then necessary to provide some interpretability.

Let us introduce briefly the most prominent representatives of these methods. First, global explanation methods such as partial dependence plots (Friedman, 2001) or Sobol indices (Sobol, 2001), quantify the main effects and the interaction effects of the explanatory variables on the dependent variable. Then, some methods simulate a black box model with a simpler, more interpretable one, through distilled additive explanations (Tan et al., 2018) or subspace explanations (Lakkaraju et al., 2019). Also, local explanation methods, such as LIME (Ribeiro et al., 2016) or SHAP (Lundberg and Lee, 2017), focus on learning simple local approximations to explain individual predictions.

The machine learning research community offers nuanced perspectives about the merits of post-hoc explanations. As a representative example, (Lipton, 2018) suggests that post-hoc explanations should not be ruled-out as valid, although indirect, means of knowledge about the underlying data generating process. He also underlines the potential risk of focusing on misleading information when relying on post-hoc explanations. Moreover, he considers that transparent linear models may not always be more interpretable than deep neural networks (DNN) because they often need heavily engineered features to obtain similar performances. Likewise, (Poursabzi-Sangdeh et al., 2021) observe that practitioners can be affected by the *information overload* phenomenon when the number of features becomes too large. Otherwise, (Rudin, 2019) emphasizes the importance of taking into account the whole data analysis process, including the preprocessing steps: "when considering problems that have structured data with meaningful features, there is often no significant difference in performance between more complex classifiers (DNN, boosted decision trees, random forests) and much simpler classifiers (logistic regression, decision lists) after preprocessing". She points out that there is not necessarily a trade-off between accuracy and interpretability: performance gaps can be reduced iteratively through better data processing and model understanding. The latter is facilitated by the use of interpretable models.

In our work, we focus on predictive models used to inform high stake decision making processes. Therefore, we consider that effective parametric models with simple functional forms are more desirable than post-hoc explanations of black box models since they directly provide the marginal effects of the explanatory variables.

2.2. Crash Prediction Models and their interpretability

Many methods have been proposed for crash frequency analysis (Lord and Mannering, 2010). The parametric statistical models, mostly represented by GLM, explicitly associate the crash related variable to a vector of explanatory variables. Crash frequencies being positive integers, they have originally been modelled by Poisson

regressions (Jones et al., 1991; Joshua and Garber, 1990; Miaou, 1994). However, an over-dispersion phenomenon is often observed in highway safety studies. Therefore, the more flexible Poisson-gamma models, also called negative binomial models, being able to adjust the variance, are often preferred to Poisson regression for crash prediction (Miaou and Lord, 2003; Lord et al., 2005; El-Basyouny and Sayed, 2006; Lord and Kuo, 2012). With the above models, roadway safety experts can understand the risk factors through the analysis of a few parameters associated with uncertainty estimates.

CPM can also be black box models, such as SVM (Li et al., 2008) or artificial neural networks (Zeng et al., 2016; Chang, 2005). With their ability to model non-linear relationships, they often have better predictive performance than Poisson-gamma models. They have been supplemented by sensitivity analysis methods to give access to an estimate of the relationship between observed variables and crash related ones (Li et al., 2008; Yu and Abdel-Aty, 2013). Nevertheless, these methods (e.g. partial dependence plots) assume independence between variables and may lead to skewed interpretations in presence of multicollinearities (Molnar, 2020). (Khoda Bakhshi and Ahmed, 2021) compare four explanation tools, including partial dependence plots (PDP), individual conditional expectation (ICE), centered ICE, and accumulated local effects (ALE). They validate that PDP should not be the unique explanation method and must be accompanied by ICE. For highly correlated spaces they also indicate that ALE plots should be endorsed. More recently, several studies (Mihaita et al., 2019; Parsa et al., 2020) use SHAP (Lundberg and Lee, 2017) to identify influencing factors and their interactions for incident duration prediction and real-time accident detection.

Finally, parametric models such as GLM can be refined into multilevel models to account for correlated responses within clusters (Jones and Jørgensen, 2003; Kim et al., 2007; Huang and Abdel-Aty, 2010). For instance, crashes occurring in a given geographical region may possess specific characteristics while not differing entirely from crashes in other regions. Ignoring this may produce misspecified and poorly estimated models (Jones and Jørgensen, 2003). In (Veran et al., 2020), we automated the discovery of such a hierarchical structure by analyzing the results of the SHAP post-hoc explanations of a highly flexible black box machine learning model. In this paper, we further exploit the discovery of this hierarchical structure by using symbolic regression to capture sparser or more complex but still interpretable relationships between the explanatory variables and the crash count.

2.3. Symbolic Regression

Symbolic regression (SR) consists in exploring a large space of functional forms to discover a predictive model with a good trade-off between accuracy and simplicity. Each element of this space is a parametric regression model whose performance is

measured (e.g., with cross-validation) on a given dataset after fitting its parameters. Thus, both the parameters and the functional form of a predictive model are learned based on available data. A wide variety of approaches have been tried to effectively explore the space of functional forms. Many of them are based on genetic programming (GP) (McKay et al., 1995; Augusto and Barbosa, 2000; Schmidt and Lipson, 2009; Haeri et al., 2017; Burlacu et al., 2020; La Cava et al., 2021) where a population of mathematical expressions evolves through selection, crossover and mutation to improve a fitness function. As another example, the metaheuristic algorithm of Pareto simulated annealing (Stinstra et al., 2008) has been used to discover a set of models which are optimal in terms of a balance of both accuracy and simplicity metrics. Thanks to the use of Meijer G-functions, SR can also be approached by algorithms based on gradient descent (Alaa and van der Schaar, 2019). Bayesian processes, with algorithms based on the Markov Chain Monte Carlo strategy have been used as well to solve the SR problem (Jin et al., 2019).

Finally, recent studies apply deep learning methods to symbolic regression. SR based on neural networks (Udrescu and Tegmark, 2020) can discover hidden simplicity in the data (e.g. symmetry, separability) to decompose complex problems into simpler sub-problems. Lately, this approach has been improved with the integration of an information complexity metric by means of Pareto optimization (Udrescu et al., 2020). (Petersen et al., 2019) use a hybrid approach that combines genetic algorithms and a recurrent neural network (RNN) trained by reinforcement learning to generate better symbolic models at each iteration. Finally, (Valipour et al., 2021) consider the problem as a sub task of language modelling and train a generative RNN model with reinforcement learning to produce symbolic equation skeletons whose constants are further adjusted by the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm (Fletcher, 2013).

SR finds applications in numerous domains such as physics (Schmidt and Lipson, 2009), finance (Chen, 2012), climate modeling (Stanislawska et al., 2012) or renewable energies (La Cava et al., 2016). So far, only few studies applied symbolic regression to safety analysis. (Meier et al., 2014) use prioritized grammar enumeration, a dynamic programming version of symbolic regression, to predict crash severity a few milliseconds before collision. (Patelli et al., 2020) design a GP-based symbolic regression to predict the traffic flow. To the best of our knowledge, symbolic regression has not been applied to long term crash predictions.

3. Interpretable hierarchical symbolic regression

3.1. Overview

In this section, we introduce our proposed framework, made of three modules depicted in Fig. 1. The *hierarchical structure* module clusters the dataset under

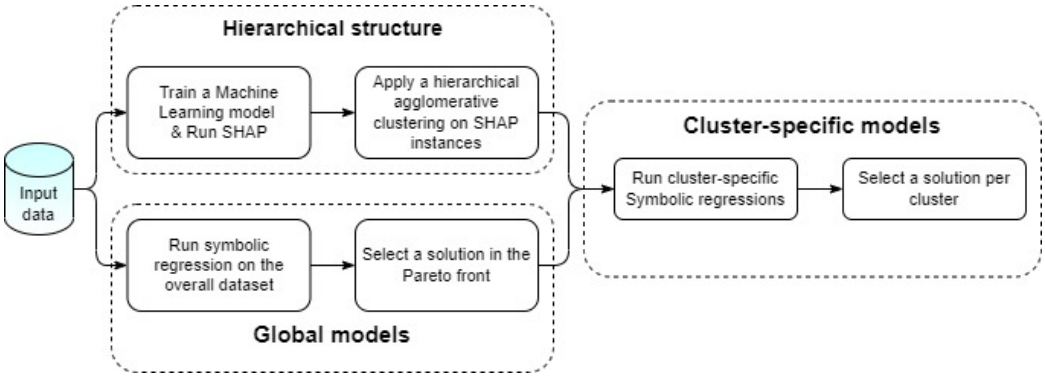


Figure 1: Proposed framework for an interpretable hierarchical symbolic regression

knowledge of the dependent variable. To do this, we draw from an analysis of a flexible black box model with the SHAP (Lundberg and Lee, 2017) explanatory framework. Then, to the whole training dataset and to each cluster, we apply a variant of the symbolic regression (SR) method to find expansions of linear models with effective and interpretable functional forms. To this end, we designed a multi-objective simulated annealing algorithm to solve the SR problem. Thus, we discover Pareto optimal predictive models with various trade-offs between accuracy and complexity. In our case, symbolic regression serves two purposes. First, it discovers *global models*, learned from the whole training dataset, that capture the associations between the explanatory variables and the target. Second, based on the hierarchical structure of the instances, our SR-based algorithm implements a partial pooling strategy to refine a global model into *cluster-specific* ones.

3.2. Supervised learning of a hierarchical structure

As in our previous work (Veran et al., 2020), we first train a state-of-the-art black box model. We select `XGBoost`[1] (Chen and Guestrin, 2016), a gradient boosting tree model known for its robustness, computational efficiency and high accuracy on tabular datasets (Chen and Guestrin, 2016; Borisov et al., 2021). Then, we apply a SHAP (Lundberg and Lee, 2017) analysis to quantify the contribution of each original explanatory variable to each individual prediction. To each observation, SHAP associates a linear function g:

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i \tag{1}$$

[1] `https://xgboost.readthedocs.io/en/stable/`

Figure 2: Forceplot of the explanatory variables' contributions to the estimated crash count for a specific observation. On this example, the traffic has the biggest positive contribution and explains most of the crash count shift from its overall expected value.

where M is the number of simplified features $z'_i \in \{0,1\}^M$ and $\phi_i \in \mathbb{R}$ are their contributions which correspond to the game theoretic concept of Shapley values.

Among the different implementations of SHAP, we select TreeSHAP, an efficient tree-based algorithm for fast and consistent computations of exact Shapley values (Lundberg et al., 2018, 2020). TreeSHAP accounts for feature dependence and also reduces the complexity of Shapley value computation from exponential to low order polynomial time when compared to kernelSHAP, the initial model-agnostic implementation of SHAP (Lundberg et al., 2018). Moreover, (Lundberg et al., 2020) observed that TreeSHAP consistently outperforms alternative methods across a benchmark of 21 different local explanation metrics.

Furthermore, in (Lundberg and Lee, 2017), the authors also propose a *forceplot* visualization (see Fig. 2) to materialize how much each contribution shifts the output relatively to the overall expected value of the black box model.

When applied to all observations, the SHAP forceplots can be clustered by similarities of their profiles. For example, on the highway network dataset, we discover clusters of roadway segments which are similar based on how the black box model transforms the original explanatory variables into an accident count. More precisely, we do a hierarchical agglomerative clustering of the observations based on the explanatory variables' contributions as provided by the SHAP analysis. We use the Ward linkage criteria and the optimal number of clusters is set by detecting the greatest increase in the squared Euclidean distance between clusters (Thorndike, 1953). We also train a decision tree classifier to learn how to associate a new observation to a cluster. In Section 4, we report on cross-validation measures showing, on various datasets, that this association is very accurate.

3.3. Symbolic Regression with Pareto simulated annealing

3.3.1. General description

Most versions of symbolic regression (SR) discover an expansion of a linear model with the addition of non-linear effects by searching a space of functional forms. Section 2.3 gave an overview of the various methods that have been used to perform this search. Among them, we select simulated annealing, an effective metaheuristic

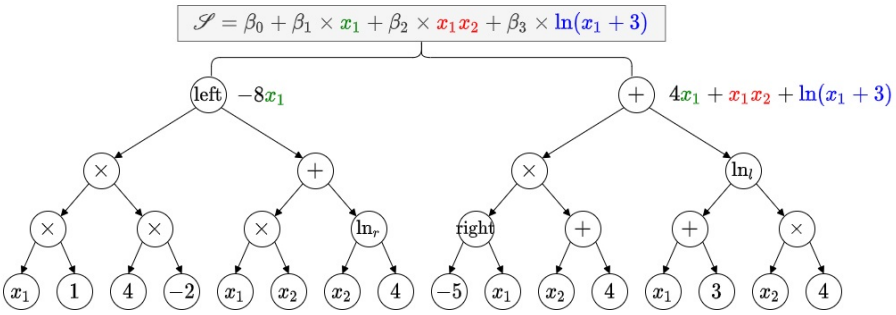


Figure 3: A functional form associated with a set of expression trees.

known for its robustness in optimization problems involving a large search space (Eren et al., 2017; Delahaye et al., 2019). Thus, we represent the problem as a local search. Moreover, we adopt a multi-objective extension of the simulated annealing algorithm to perform the search while optimizing both the complexity and the accuracy of the models (Stinstra et al., 2008). The search ends on a set of mutually non-dominated predictive models, the Pareto front.

3.3.2. Definition of a solution

The functional form of a model is extracted from a set of expression trees. Each expression tree is perfect, binary and consists of internal operator nodes and leaves. Leaves are either represented by a constant or an explanatory variable. Operator nodes can be unary (e.g., *cos*, *sin*, *tan*, *exp*, *ln*, *left*, *right*) or binary (e.g., $+, \times, /$) and have two children. For unary operators, we indicate with the subscripts " l " (for "left") and " r " (for "right") to which child the operation is applied. For instance, if the operator is ln_l, then the logarithm is applied to the left child. The *left* and *right* operators apply the identity function to the left and right child, respectively.

We extract the symbolic expression by a breadth-first traversal of the expression tree. In practice, as operators, constants and input variables are defined with `Sympy`, an open-source Python library for symbolic computation (Meurer et al., 2017), the traversal returns a `Sympy` expression. Finally, the functional form $\mathscr{S}$ of a solution is obtained by the combination and algebraic simplification of the `Sympy` symbolic expressions of a set of expression trees (see Fig. 3).

3.3.3. Initialization of a first solution

Function `initialize` of algorithm 1 generates a first solution represented by a set $\mathscr{M}_{cur}$ of random expression trees, with $\mathscr{S}_{cur}$ the associated functional form. To this end, this function first creates balanced binary trees, each of the same depth. Then, for each tree, an inorder traversal associates an index to each node. Odd indices refer to internal nodes while even indices refer to the leaves (see Fig. 4a). In this way, we

have an efficient means to search for a node in a tree and to know directly what type of node it is (see Section 3.3.4). At the same time, internal nodes are initialized with an operator chosen with equiprobability from the set of predefined operators introduced in Section 3.3.2. Each leaf has a 50% probability of being initialized either to a constant or to one of the explanatory variables. In the latter case, each explanatory variable is equiprobable.

Algorithm 1 Symbolic regression with Pareto simulated annealing

Require: m: number of expression trees; $T_{min} = 0.0001$: initial temperature (heating phase) and minimum temperature (cooling phase); $\lambda_h = 1.15, \lambda_c = 0.85$: ratios between two adjacent temperatures in the heating phase and cooling phase, respectively; $s_h = s_c = 300$: number of iterations between two updates of temperature; $\gamma_c = 1.15$: ratio that controls the growth of s_c; max: maximum number of iterations during the cooling phase.

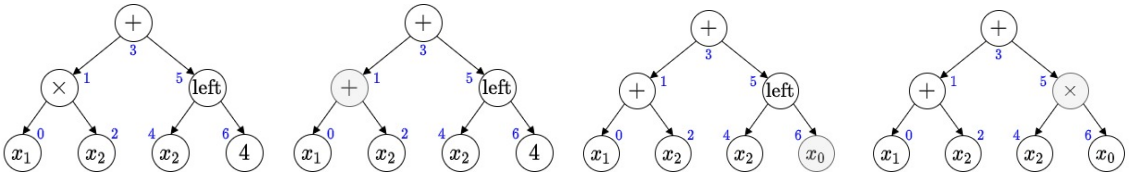
function SIMULATED ANNEALING($T_{min}, \lambda_h, s_h, \lambda_c, s_c, \gamma_c, max$)
  $T = T_{min}$ ▷ Annealing temperature
  $\zeta = \varnothing$ ▷ Pareto front
  $acc = 0, rej = 0$ ▷ Number of accepted and rejected solutions
  $\alpha = 0$ ▷ Acceptance rate
  $i = 0$
  $\mathscr{M}_{curr}, \mathscr{S}_{curr} \leftarrow$ `initialize` (m)
  while $\alpha \leq 0.9$ **do** ▷ Heating phase
    $i, \zeta, \mathscr{M}_{curr}, \mathscr{S}_{curr}, acc, rej \leftarrow$ `explore`$(T, i, \zeta, \mathscr{M}_{curr}, \mathscr{S}_{curr}, acc, rej)$
    if $i \bmod s_h = 0$ **then**
      $T \leftarrow T \times \lambda_h$, $\alpha \leftarrow acc/(acc + rej)$
      $acc \leftarrow 0, rej \leftarrow 0$
  $i = 0, \zeta = \varnothing$
  $\mathscr{M}_{curr}, \mathscr{S}_{curr} \leftarrow$ `initialize` (m)
  while $T > T_{min}$ and $i < max$ **do** ▷ Cooling phase
    $i, \zeta, \mathscr{M}_{curr}, \mathscr{S}_{curr}, acc, rej \leftarrow$ `explore`$(T, i, \zeta, \mathscr{M}_{curr}, \mathscr{S}_{curr}, acc, rej)$
    if $i \bmod s_c = 0$ **then**
      $T \leftarrow T \times \lambda_c$, $s_c \leftarrow s_c \times \gamma_c$
  return ζ

3.3.4. Neighbourhood of a solution

Function `generate` of algorithm 2 generates a new solution $\mathscr{S}_{new}$ in the neighbourhood of the current solution $\mathscr{S}_{cur}$. It randomly selects an expression tree from $\mathscr{M}_{cur}$ and a node index from $\{0, ..., 2^T - 2\}$, T being the tree depth. Then, a recursive

search finds the node with the selected index. When the node is an operator (*viz.*, its index is odd), it is replaced by a randomly selected operator. Likewise, when a leaf is selected (*viz.*, its index is even), it is replaced by a randomly selected constant or explanatory variable. Fig. 4 illustrates this process.

If unchecked, function `generate` could lead to ill-defined operators. For instance, a logarithm could be applied to a potentially negative domain. Therefore, function `integrityCheck` infers recursively the domain of each operator node and, based on rules from interval arithmetic, checks its validity (Table 1 introduces some of these rules). With interval arithmetic, we have an efficient way to ensure that the functional form generated from the random process does not contain any undefined values (Keijzer, 2003). For more details on the integrity check, we refer to (Stinstra et al., 2008).



(a) $x_1x_2 + x_2$ (b) $x_1 + 2x_2$ (c) $x_1 + 2x_2$ (d) $x_1 + x_2 + x_0x_2$

Figure 4: A sequence of transformations applied to an expression tree. (a) An initial expression tree. Blue integers refer to node indices. (b) A transformation is applied to operator node 1, thus modifying the underlying functional form. (c) The transformation applied to leaf node 6 is muted due to its *left* operator parent. (d) Later in the process, node 6 can be reactivated when its parent is transformed.

Operation	Lower bound	Upper bound	Invalid if
$[a,b] + [c,d]$	$a + c$	$b + d$	
$[a,b] - [c,d]$	$a - d$	$b - c$	
$[a,b] \times [c,d]$	$\min\{ac, ad, bc, bd\}$	$\max\{ac, ad, bc, bd\}$	
$[a,b]/[c,d]$	$\min\{a/c, a/d, b/c, b/d\}$	$\max\{a/c, a/d, b/c, b/d\}$	$0 \in [c,d]$
$left([a,b],[c,d])$	a	b	
$ln_l([a,b],[c,d])$ [a]	$\ln(a)$	$\ln(b)$	$a \leq 0$

[a] ln_l designates a ln operator applied to the left child

Table 1: Rules for interval arithmetic, from (Stinstra et al., 2008, p.318). We suppose that an operator node has two children. The left one is defined on $[a,b]$ and the right one on $[c,d]$.

3.3.5. Cost of a solution

The cost of a solution is measured in terms of both the prediction error (see function `measurePerformance` of algorithm 2) and the complexity of the functional

Algorithm 2 Pseudo code of function `explore`

```
1: function explore(T, i, ζ, M_curr, S_curr, acc, rej)
2:     M_new ← generate(M_curr)
3:     if integrityCheck(M_new) then
4:         S_new ← simplify(M_new)
5:         if S_new ≠ S_curr then
6:             perf_new ← measurePerformance(S_new)
7:             compl_new ← measureComplexity (S_new)
8:             if accept(perf_new, compl_new, perf_curr, compl_curr, ζ, T) then
9:                 ζ, M_curr, S_curr, perf_curr, compl_curr ← update(
                       ζ, M_new, S_new, perf_new, compl_new)
10:                acc ← acc + 1
11:            else
12:                rej ← rej + 1
13:        else
14:            M_curr ← M_new
15:        i ← i + 1
16:    return i, ζ, M_curr, S_curr, acc, rej
```

form (see function `measureComplexity` of algorithm 2).

To obtain a robust estimate of the prediction error of $\mathscr{S}_{new}$, we compute the average RMSE on the validation subsets of a 5-fold cross-validation process. On each training subset, the coefficients β_i of $\mathscr{S}_{new}$ are learned by solving an $l2$-regularized linear regression. The regularization parameter is determined on each training subset of the 5-fold cross-validation by an efficient generalized cross-validation (Golub et al., 1979). Once the estimate of the prediction error is obtained, the coefficients β_i are fitted one last time on the whole training dataset.

We improve the strategy introduced in (Stinstra et al., 2008) to propose a new measure of the complexity of a solution. We penalize both the collinearities and the number of terms present in the symbolic expression of the functional form. The complexity of a solution $\mathscr{S}$ composed of m terms is defined as:

$$\text{Complexity}(\mathscr{S}) = \sum_{i=1}^{m} \Big(1 + \max\big(\{|r_{ij}|; j \in \{1, 2, ..., m\} \setminus i\}\big)\Big) C_i \tag{2}$$

where r_{ij} is the Pearson's correlation coefficient, computed on the training dataset, between terms i and j, and C_i is the complexity of the term i. We use algebraic rules to compute the complexity of each term, some of which are presented in Table 2. The complexity of a unary operator (e.g., the natural logarithm) is determined by

Term i	Complexity C_i	Example	Computed C_i
const	0	2	0
x	1	x_4	1
$f(x)^n$	$n \times C(f(x))$ [a]	x_2^2	2
$f(x) \times g(y)$	$C(f(x)) + C(g(y))$	$x_1 x_2^2$	3
$f(g(y))$	$C(f(x)) \times C(g(y))$	$\ln(3x_2^2)$	$C_{unary}(\ln) \times 2$ [b]

[a] $C(.)$ is the complexity of the inner function

[b] $C_{unary}(.)$ is the complexity of the unary operator

Table 2: Algebraic rules used to compute the complexity of each term, adapted from (Stinstra et al., 2008, p.320)

approximating the operator, on its inferred domain, by a polynomial of increasing degree (at most 10) until the score of the fit, as measured on a validation set, is below a predefined threshold. The complexity of the unary operator is then defined as the degree of the best polynomial approximation. It should also be noted that, according to equation 2, the more terms a solution has, the more complex it is. We were able to confirm experimentally that the measured complexity represents well the complexity perceived by the safety experts.

3.3.6. Comparison of two solutions

The search ends with a set of Pareto optimal solutions that belong to the boundary beyond which neither the prediction error nor the complexity can be improved without deteriorating the other objective. This can be formally defined in terms of a dominance relation. Let U_1 be the prediction error and U_2 be the complexity metric.

$$\begin{aligned} & \mathscr{S}_a \text{ dominates } \mathscr{S}_b \\ \equiv \\ & \forall i \in \{1,2\} \ : \ U_i(\mathscr{S}_a) \leq U_i(\mathscr{S}_b) \quad \text{and} \quad \exists j \in \{1,2\} \text{ s.t. } U_j(\mathscr{S}_a) < U_j(\mathscr{S}_b) \end{aligned}$$

Thus, the search returns a set of non-dominated solutions called the Pareto front.

3.3.7. Exploration by Pareto simulated annealing

Simulated annealing (SA) is an iterative local search process used to solve optimization problems for which a simple hill-climbing approach would most often converge on a poor local optimum. At each iteration, SA generates randomly a solution $\mathscr{S}_{new}$ in the neighborhood of the current solution $\mathscr{S}_{cur}$. The probability P of accepting $\mathscr{S}_{new}$ as the new current solution is a function of both a temperature parameter T and the difference in cost ΔE between the two solutions.

$$P = e^{-\Delta E/T} \tag{3}$$

Annealing Temperature T. SA mimics the physical process of annealing in metallurgy where a material is first heated before being gradually cooled in order to reach an equilibrium state with increased ductility and hardness. SA follows a similar two-steps process.

The heating phase aims at discovering an initial temperature T_0 that favors exploration over exploitation in the beginning of the search. The heating process starts from a low temperature at which a deteriorating neighbour of the current solution is rarely accepted. Then, every s_h iterations, the temperature is increased according to a geometric series of ratio $\lambda_h > 1$. The process ends at a temperature T_0 at which at least 90% of the randomly generated neighbours are accepted.

During the cooling phase, the annealing temperature is progressively decreased, every s_c iterations, according to a geometric series of ratio $\lambda_c < 1$. High temperatures favor the exploration of the space of functional forms by preventing the process from converging too early on a local optimum. On the contrary, the more the temperature decreases, the less likely it is for a deteriorating neighbour to replace the current solution. The value of λ_c controls the speed at which the annealing temperature decreases. If λ_c is too small, the optimization may stay stuck too early in the neighborhood of a poor local optimum. Whereas, if λ_c is too close to 1, the optimization may take too long to reach a good optimum. Moreover, parameter s_c increases according to a geometric series of ratio $\gamma_c > 1$. Thus, more iterations are allocated to lower temperatures to favor the exploitation of promising functional forms. Finally, the search ends when either the temperature falls below a threshold or the number of iterations reaches a predefined maximum.

ΔE and the acceptance of a new solution. For a single-objective optimization problem, ΔE is simply the difference of the objective function evaluated at two neighbouring solutions. For our multi-objective optimization problem, we use the dominance-based performance metric introduced above. When a new solution $\mathscr{S}_{new}$ dominates, or is as good as, $\mathscr{S}_{cur}$, it is accepted as the new solution (see function `accept` of algorithm 3). When $\mathscr{S}_{new}$ is less effective than $\mathscr{S}_{cur}$, it has a probability P defined by eq. 3 to be accepted. In that case, ΔE is defined as:

$$\Delta E(\mathscr{S}_{cur}, \mathscr{S}_{new}) = \frac{1}{|\tilde{\zeta}|}\Big(|\tilde{\zeta}_{\mathscr{S}_{new}}| - |\tilde{\zeta}_{\mathscr{S}_{cur}}|\Big) \tag{4}$$

with ζ the set of solutions that approximate the Pareto front, $|\tilde{\zeta}|$ the cardinality of $\zeta \cup \{\mathscr{S}_{cur}, \mathscr{S}_{new}\}$, and $|\tilde{\zeta}_{\mathscr{S}}|$ the number of solutions in $|\tilde{\zeta}|$ that dominate $\mathscr{S}$ (see Fig. 5). Moreover, to smooth the estimated acceptance probability distribution, new artificial points are added to the attainment surface to get an evenly spread attainment surface over the two dimensions of the Pareto front (Smith et al., 2004).

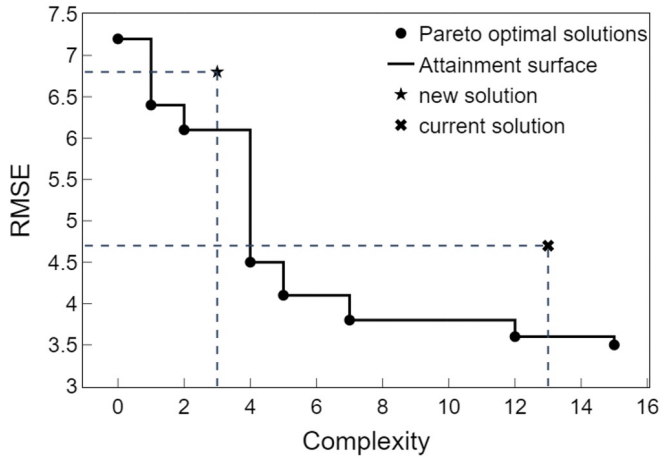


Figure 5: Example of an approximated Pareto front and its attainment surface, adapted from (Stinstra et al., 2008, p. 322). From Eq. 4, $\Delta E(\mathscr{S}_{cur}, \mathscr{S}_{new}) = (2-4)/9 = -2/9$

Finally, when $\mathscr{S}_{new}$ is accepted, the Pareto front ζ is updated (see function `update` of algorithm 3) by removing the solutions dominated by $\mathscr{S}_{new}$ and then adding $\mathscr{S}_{new}$ to ζ when it is not dominated by any other solution in ζ. Thus, at the end of each iteration, ζ is the set of non-dominated solutions encountered during the search.

3.4. Automatic selection of a global model

In Section 5, where we illustrate the dynamic interpretative process made possible by our framework, we emphasize the usefulness of being able to let the user choose a predictive model on the Pareto front. In that way, the end user can precisely balance between the predictive performance and the simplicity of the model. However, in our proposed methodology, we also need a principled way to automatically select a model on the Pareto front. To do this, first, we consider the point Ω in the Pareto plan that, (i) on the performance axis, is at the level of the most efficient model encountered and, (ii) on the complexity axis, is at the level of the simplest model encountered. Then, we select the model $\mathscr{S}_{glob}$ on the Pareto front closest to Ω in the sense of the Euclidean distance. This model, located in the elbow of the Pareto front, is likely to offer a good trade-off between predictive performance and complexity. In the next stage of our approach, it is used as a starting point to build cluster-specific models.

3.5. Cluster-specific models

To discover cluster-specific phenomena, for each cluster discovered by the approach introduced in Section 3.2, a modified version of the symbolic regression search is conducted. It consists in merging the functional form built from the expression trees with the fixed functional form of $\mathscr{S}_{glob}$ (see Fig. 6): common terms are grouped

Algorithm 3 Pseudocode of `accept` and `update` functions

function `accept`($\text{perf}_{new}, \text{compl}_{new}, \text{perf}_{curr}, \text{compl}_{curr}, \zeta, T$)
    $is\_accepted \leftarrow$ `False`
    if $\text{perf}_{new} \leq \text{perf}_{curr}$ and $\text{compl}_{new} \leq \text{compl}_{curr}$ **then**
        $is\_accepted \leftarrow$ `True` ▷ new solution dominates, or is as good as, the current one
    else
        compute P according to Eq. 3
        draw randomly j in $[0, 1]$
        if $P \geq j$ **then**
            $is\_accepted \leftarrow$ `True`
    return $is\_accepted$

function `update`($\zeta, \mathscr{M}_{new}, \mathscr{S}_{new}, \text{perf}_{new}, \text{compl}_{new}$)
    $is\_dominated =$ `False`
    for solution $\mathscr{S}$ in the Pareto front ζ **do**
        if $\mathscr{S}$ dominates $\mathscr{S}_{new}$ **then**
            $is\_dominated =$ `True`
    if $is\_dominated =$ `False` **then**
        remove solutions in ζ dominated by $\mathscr{S}_{new}$
        add $\mathscr{S}_{new}$ to ζ
    return $\zeta, \mathscr{M}_{new}, \mathscr{S}_{new}, \text{perf}_{new}, \text{compl}_{new}$

together and new terms are added to the formula. Hence, the marginal effects already represented by $\mathscr{S}_{glob}$ can be reduced or amplified and new cluster-specific effects can be discovered. It corresponds to a partial pooling approach where cluster-specific models can benefit from the effects already discovered by the global model.

3.6. Uncertainty estimation

Our approach results in global and cluster-specific expansions of linear models. Therefore, the marginal effects of the terms composing the models are readily interpretable. However, since the training set has been used to estimate the $l2$-regularization hyper-parameters, there is no simple linear relationship between uncertainty in the parameters and uncertainty in the target. Bootstrap techniques could estimate the uncertainty in the parameters. Still, a standard bootstrap approach is not appropriate since the bias introduced by the penalty term would not be correctly estimated. Double bootstrap techniques have been proposed (Vinod, 1995; McCullough and Vinod, 1998) to take into account an estimation of the bias. Nonetheless, they are computationally expensive ($O(n^3)$ where n is the number of samples). Also,

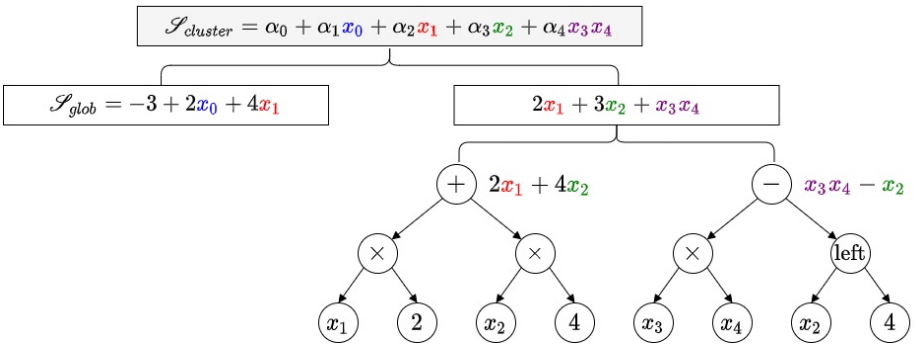


Figure 6: Extraction of a cluster-specific functional form from a set of expression trees and the fixed functional form of the global model

asymptotic statistics have been derived to measure the uncertainty in the parameters under a fixed setting of the regularization parameter (Firinguetti and Bobadilla, 2011). They wouldn't be appropriate in our case since we estimate the regularization parameter by leave-one-out cross-validation. Therefore, we make use of a well-known equivalence (Mehta et al., 2019) between the ridge regression regularization parameter and the parameters of a Gaussian prior for the Bayesian formulation of linear regression. It can be shown that the variance τ^2 of the zero-centered Gaussian prior must be defined as:

$$\tau^2 \equiv \frac{\sigma^2}{\lambda}$$

where λ is the ridge regularization parameter and σ^2 is the variance of the likelihood that can be estimated by measuring the variance of the target on the training dataset. For each Pareto optimal solution, we start from the discovered functional form and the value of the ridge regularization hyper-parameter λ to infer again the coefficients, but this time, using Bayesian inference with the above prior. The resulting posterior distributions give an estimate of the parameters' uncertainty.

4. Experiments

4.1. Datasets and preprocessing

We test our framework on a highway network dataset (further referred to as *French Highway*). The task is to predict yearly crash counts on 10 km long highway segments of a french highway network (2300 km). The predictors include topographical data (number of lanes, right shoulder width...), average annual daily traffic (AADT), speed limits and average altitudes. The dataset covers 11 years of data, from January 1^{st}, 2008 to December 31^{th}, 2018.

Dataset	#Instances	#Features	dependent variable
French Highway	4152	11	crash count
Insurance (Lantz, 2013)	1338	7	health insurance costs
Airbnb (Airbnb, 2019)	48895	12	housing prices
House (218_house_8L)*	22784	8	-
Puma (225_puma8NH)*	8192	8	-
Satellite (294_satellite_image)*	6435	36	-
Wind (503_wind)*	6574	14	-
Breast tumor (1201_BNG_breastTumor)*	116640	9	-
Music (4544_GeographicalOriginalofMusic)*	1059	117	-
Wine†	4898	12	white wine quality
Toxicity†	546	9	aquatic toxicity
Gas†	36733	11	gas emission

* Datasets taken from `https://epistasislab.github.io/pmlb/index.html`

† Datasets taken from `https://archive.ics.uci.edu/ml/index.php`

Table 3: Datasets

We also test our framework on 11 public regression datasets (see Table 3) from different domains and with various volumes (from 546 to 116640 instances) and dimensionalities (from 7 to 117 features).

For preprocessing, categorical variables are one-hot-encoded and continuous variables are standardized.

4.2. Performance metric

To measure the performance of predictive models, we use the Root Mean Square Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$$

with n the number of observations, y_i the target and $\hat{y}_i$ the predicted value.

4.3. Implementation details

We implement our model in `Python`. In order to converge towards interpretable models, we restrict the operators available to the symbolic regression to $\{left, right, ln\}$ for the unary ones, and $\{\times, +, -\}$ for the binary ones. For the algebraic simplification of the expression trees by, e.g., grouping common terms together (see function `simplify` in algorithm 2), we use a module[2] from the `Sympy` library.

[2] `https://docs.sympy.org/latest/modules/simplify/simplify.html`

To fit the coefficients of a newly discovered functional form, we use the `scikit-learn`[3] implementations of ridge regression. The optimal coefficients of the linear models are computed with $l2$-regularized least squares. Indeed, with the introduction of a weight decay, better generalization performances can usually be achieved and the models are less prone to the negative effects of multicollinearities. We optimize the $l2$-regularization parameter by an efficient form of leave-one-out cross-validation (viz., generalized cross validation).

As explained in Section 3.6, in order to endow our final models with uncertainty estimates, we use Bayesian inference to compute the posteriors for the coefficients of each functional form on the Pareto front. We apply gaussian priors corresponding to the already known optimal value of the regularization hyper-parameter (see Section 3.6). We rely on the `pymc3`[4] library with the *No U-Turn Sampler* (Hoffman et al., 2014) to run simultaneously two Markov chains for 3000 iterations, with a burn-in period of 1000 iterations.

4.4. Preliminary search for Symbolic Regression's hyper-parameters

On the *French Highway* and *Insurance* datasets, we conduct a grid-search for the symbolic regression's hyper-parameters (viz. the number of expression trees and the depth of a tree). We observe that deeper trees lead to more complex models. This is mainly due to the possibility of deep compositions of functions. Thus, motivated by finding a good compromise between the complexity of the final models and their predictive performance, we restrict the tree depth to 4. In this way, each expression tree, being a perfect binary tree, has 8 leaves. Furthermore, the grid-search also reveals that there is no noticeable improvement in predictive performance when the number of expression trees exceeds two-thirds of the number of features after data preparation (viz., one hot encoding). The number of trees used for each dataset is given in Table 5.

4.5. Models used for comparison

We compare our proposal to models of varying degrees of interpretability. Among the most simple and interpretable models, we select the `scikit-learn` implementations of ordinary least square regression (*OLS*) and *decision trees* of depth no more than 5 (to preserve interpretability).

We also compare our current approach to a previous proposal of ours, a Bayesian hierarchical GLM (*BH-GLM*) (Veran et al., 2020). The latter is based on the bayesian inference of a linear hierarchical model with data-driven discovery of objective priors in the form of i) a hierarchical structure and ii) strong first-order interactions.

[3] `https://scikit-learn.org/stable/`
[4] `https://docs.pymc.io/en/v3/`

The hierarchical structure is learned by the same method as the one introduced in Section 3.2. The retained strong first-order interactions are obtained through the analysis of the structure of a trained self-adaptive polynomial network. We also consider *B-GLM*, a variant with first-order interactions that does not account for the hierarchical structure.

Moreover, we include two variants of Generalized Additive Models (GAM). For the first one, *GAM-splines*, based on a spline basis, we use the `PyGAM`[5] implementation. For the second one, explainable boosting machine (*EBM*), based on gradient boosting with bagging, we use the implementation provided by the `InterpretML` framework (Nori et al., 2019).

We also compare our approach to genetic programming based symbolic regressions with the reference implementations of the `gplearn`[6] package (*SR-GP*) and GP-GOMEA[7] (*SR-Gomea*) (Virgolin et al., 2021), the latter being known to perform well on many real world datasets (La Cava et al., 2021). For both implementations, the set of operators is restricted to the one we use in our approach (*viz.*, $\{+, -, \times, ln\}$). We also consider *SR-Gomea-op*, the same model with a less restricted set of operators (*viz.*, $\{+, -, \times, ln, cos, sin, \sqrt{}\}$), the same as the one used by (La Cava et al., 2021) in their recent survey.

For all the aforementioned interpretable models, we apply a no pooling approach that accounts for the clusters discovered by our *Hierarchical structure* module (see Section 3.2). This approach fits a separate model for each cluster and considers that no similarities exist between them.

Finally, we select three highly flexible black box models: (i) the `scikit-learn` implementation of Support Vector Machines (SVM) and (ii) Multilayer Perceptrons (MLP), and (iii) the `XGBoost` gradient tree boosting library.

For fair comparisons, the hyper-parameters of the aforementioned models are optimized by cross-validation with grid-search. For each model, the grid of hyper-parameters' values are given in Appendix A.

For the experiments, we consider several variants of our framework. *SR-trad* and *SR-max* use only the global model of Section 3.4 while *HSR-trad* and *HSR-max* use the cluster-specific models of Section 3.5 (prefix "H" stands for *hierarchical*). We also train our hierarchical symbolic regression on clusters discovered with a hierarchical agglomerating clustering applied to the training data, including the dependent variable. These models are further referred to as *HSR-naive-trad* and *HSR-naive-max*. Finally, *SR-NP-trad* and *SR-NP-max* are cluster-specific models learned with a no pooling

[5] `https://pygam.readthedocs.io/en/latest/`

[6] `https://gplearn.readthedocs.io/en/stable/`

[7] `https://github.com/marcovirgolin/GP-GOMEA`

Dataset	#clusters	f1 (std)
French Highway	4	0.995 (0.003)
Insurance	2	1.0 (0.0)
Airbnb	4	0.985 (0.005)
House	4	0.908 (0.026)
Puma	3	0.975 (0.013)
Satellite	3	0.964 (0.003)
Wind	3	0.944 (0.022)
Breast tumor	2	0.995 (0.004)
Music	3	0.96 (0.029)
Wine	3	0.957 (0.012)
Toxicity	2	0.95 (0.039)
Gas	2	0.99 (0.003)

Table 4: For each dataset: number of clusters selected and performance of the prediction to associate a new observation to its cluster

approach, meaning that they do not include the knowledge of the global model. The suffixes *trad* and *max* are used to distinguish models selected near the elbow of the Pareto front, that should have a good trade-off (whence *trad*) between complexity and predictive performance, from models of maximum complexity (whence *max*).

Results, averaged from a 5-fold cross-validation, are reported in Table 5 and Table 6, the latter for cluster-specific interpretable models trained with a no pooling approach.

4.6. Hierarchical structure module

For each dataset, the optimal number of clusters computed in the *Hierarchical structure* module is given in Table 4. Moreover, to validate the ability of this module to associate an unknown sample to a cluster, a train-test split approach is applied on each training subset of the 5-fold cross-validation. For each training subset, the decision tree classifier is trained, on 80% of the data, to predict, based on the explanatory variables, the cluster to which a new observation belongs. A $f1$-score is computed on the remaining 20% of each training subset. The decision tree classifier is highly accurate on all datasets (see Table 4).

4.7. Results

HSR-trad, the model that, according to our approach, should offer a good trade-off between performance and complexity, obtains better RMSE than *OLS* and *decision tree* on all datasets but *Insurance* and *Music*. As expected, *HSR-max*, the most complex model resulting from our approach, obtains better RMSE than *HSR-trad*,

except for the *Insurance* dataset where they obtain similar predictive performance. Moreover, the fact that our models have performance metrics with low standard deviations testifies to their robustness. Indeed, they are likely to discover similar solutions on similar datasets.

Our approach based on hierarchical symbolic regression is more efficient than *B-GLM* and *BH-GLM*. This can be explained by the flexibility of our approach that captures a greater variety of sources of nonlinearities and interactions between explanatory variables.

SR-GP, the symbolic regression based on genetic programming, obtains poor results and is even dominated by the fully interpretable models on all datasets. The more recent approach *SR-Gomea* obtains better predictive performance than *SR-GP* but is still dominated by *HSR-trad* and *HSR-max*. *SR-Gomea-op* does not highlight significant predictive gains compared to *SR-Gomea* on most datasets. This validates that restricting the operators makes it possible to obtain interpretable functional forms with more than satisfactory predictive performance on real world datasets.

HSR-trad and *HSR-max*, the cluster-specific models, often show a clear improvement when compared to the global models *SR-trad* and *SR-max*. The partial pooling approach has a clear interest given that *HSR-trad* and *HSR-max* outperform *SR-NP-trad* and *SR-NP-max*, their no pooling variants. This can be explained by the fact that, in the no pooling case, models are trained independently on cluster data, which results in a higher risk of overfitting (Gelman and Hill, 2006). In our partial pooling approach, we use a global model as an initial seed in order to tend to increase the bias and decrease the variance of the final cluster-specific models, thus reducing the risk of overfitting and improving the predictive performances. Moreover, we also observe that incorporating the data-driven discovery of a hierarchical structure not only provides better predictive performances, it also offers better interpretability by capturing cluster-specific phenomena (see Section 5).

Our approach for discovering a hierarchical structure is robust, efficient and obtain better results on all datasets compared to the approach that considers more naive clusters. Indeed, on all datasets, *HSR-trad* and *HSR-max* are substantially better than *HSR-naive-trad* and *HSR-naive-max* (see Table 7).

HSR-max and *GAM-splines* have similar performances on all datasets but the *Insurance* dataset where *HSR-max* discovers a significant interaction between the body mass index and being a smoker. However, the no pooling variant of *GAM-splines* is slightly better than *HSR-max* on the *Insurance* dataset. Finally, *EBM* performs well on all datasets. It obtains the best performances on the *Insurance* and *Gas* datasets and is similar, if not better, to *XGBoost* on the *French Highway*, *Airbnb*, and *Wine* datasets.

	RMSE (std)					
	French Highway (8 [a])	*Insurance* (6)	*Airbnb* (12)	*Puma* (6)	*Satellite* (24)	*Wind* (9)
Local	**5.052 (0.250)**	-	-	-	-	-
OLS	6.213 (0.496)	6077 (287)	0.50 (0.011)	4.471 (0.068)	1.213 (0.009)	3.289 (0.104)
Decision tree	6.134 (0.422)	4739 (324)	0.490 (0.011)	3.688 (0.05)	1.061 (0.022)	3.839 (0.112)
GAM-splines	5.80 (0.344)	6021 (299)	0.464 (0.011)	4.236 (0.065)	0.90 (0.009)	3.082 (0.084)
EBM	5.363 (0.233)	**4533 (339)**	0.450 (0.013)	3.283 (0.049)	0.851 (0.035)	3.140 (0.089)
SR-GP	8.06 (0.730)	5168 (360)	0.563 (0.026)	4.504 (0.086)	1.628 (0.438)	3.858 (0.216)
SR-Gomea	6.309 (0.303)	4885 (251)	0.502 (0.010)	0 3.362 (0.033)	1.164 (0.028)	3.306 (0.128)
SR-Gomea-op	6.297 (0.267)	4815 (250)	0.514 (0.01)	3.238 (0.059)	1.102 (0.032)	3.296 (0.101)
B-GLM	6.356 (0.513)	5151 (293)	0.497 (0.01)	4.282 (0.065)	1.117 (0.044)	3.295 (0.098)
BH-GLM	6.004 (0.46)	4925 (301)	0.474 (0.02)	3.871 (0.13)	0.952 (0.029)	3.291 (0.106)
XGBoost	5.571 (0.377)	4667 (346)	**0.440 (0.011)**	3.257 (0.056)	**0.667 (0.033)**	3.084 (0.075)
SVM	6.310 (0.645)	4953 (272)	0.503 (0.011)	4.493 (0.089)	1.261 (0.028)	3.307 (0.102)
MLP	6.140 (0.462)	4867 (347)	0.463 (0.013)	**3.170 (0.052)**	0.789 (0.047)	**3.076 (0.087)**
SR-trad	6.258 (0.509)	5219 (330)	0.510 (0.018)	3.961 (0.031)	1.175 (0.05)	3.342 (0.104)
SR-max	6.186 (0.469)	4889 (293)	0.507 (0.011)	3.528 (0.042)	1.018 (0.04)	3.176 (0.081)
HSR-trad	5.921 (0.54)	4840 (308)	0.475 (0.012)	3.30 (0.084)	0.95 (0.034)	3.205 (0.074)
HSR-max	5.80 (0.507)	4844 (304)	0.470 (0.011)	3.277 (0.082)	0.934 (0.056)	3.198 (0.074)
	Breast tumor (6)	*Music* (78)	*House* (5)	*Wine* (8)	*Toxicity* (6)	*Gas* (7)
OLS	10.023 (0.036)	**0.465 (0.039)**	41563 (1270)	0.754 (0.02)	1.256 (0.097)	8.112 (0.133)
Decision tree	9.844 (0.039)	0.705 (0.066)	35752 (1199)	0.753 (0.015)	1.394 (0.12)	7.705 (0.127)
GAM-splines	9.663 (0.047)	0.898 (0.101)	33460 (1405)	0.728 (0.029)	1.245 (0.106)	5.993 (0.10)
EBM	9.519 (0.049)	0.60 (0.036)	31062 (1192)	0.689 (0.019)	1.20 (0.095)	**5.476 (0.072)**
SR-GP	10.441 (0.277)	0.71 (0.154)	56615 (21299)	0.857 (0.068)	1.457 (0.264)	10.75 (1.107)
SR-Gomea	9.988 (0.056)	0.523 (0.081)	36750 (1693)	0.742 (0.022)	1.343 (0.189)	8.703 (0.144)
SR-Gomea-op	9.973 (0.049)	0.499 (0.036)	36865 (1434)	0.739 (0.021)	1.267 (0.116)	8.742 (0.278)
B-GLM	9.995 (0.034)	0.469 (0.045)	39811 (1375)	0.751 (0.019)	1.246 (0.089)	8.112 (0.137)
BH-GLM	9.751 (0.036)	0.467 (0.037)	35039 (1820)	0.737 (0.017)	1.237 (0.098)	6.87 (0.282)
XGBoost	**9.435 (0.048)**	0.507 (0.046)	**29630 (1237)**	**0.68 (0.014)**	**1.157 (0.129)**	5.705 (0.190)
SVM	10.045 (0.036)	0.472 (0.042)	44879 (1676)	0.748 (0.018)	1.286 (0.195)	6.954 (0.372)
MLP	9.67 (0.04)	0.498 (0.042)	36004 (851)	0.757 (0.071)	1.280 (0.153)	6.023 (0.239)
SR-trad	10.096 (0.064)	0.543 (0.081)	37412 (2150)	0.744 (0.024)	1.253 (0.089)	8.153 (0.686)
SR-max	10.03 (0.079)	0.476 (0.045)	34716 (2049)	0.731 (0.019)	1.233 (0.106)	7.395 (0.553)
HSR-trad	9.727 (0.056)	0.497 (0.05)	33542 (1127)	0.724 (0.019)	1.243 (0.097)	7.181 (0.618)
HSR-max	9.662 (0.061)	0.471 (0.053)	33102 (833)	0.712 (0.014)	1.214 (0.065)	6.421 (0.150)

[a] number of trees for *SR-** and *HSR-** models

[b] averages and standard deviations of the performance metric obtained on 5-fold cross-validation

Table 5: Results obtained on 12 regression datasets

	RMSE (std)					
	French Highway	*Insurance*	*Airbnb*	*Puma*	*Satellite*	*Wind*
OLS	6.05 (0.45)	4971 (293)	0.484 (0.01)	3.867 (0.127)	1.101 (0.053)	3.291 (0.094)
GAM-splines	5.747 (0.467)	4913 (346)	0.751 (0.649)	3.523 (0.076)	0.899 (0.042)	**3.108 (0.087)**
EBM	**5.18 (0.275)**	**4512 (354)**	**0.444 (0.012)**	3.320 (0.057)	**0.874 (0.03)**	3.200 (0.055)
SR-GP	6.165 (0.555)	5795 (691)	0.512 (0.015)	3.975 (0.178)	1.209 (0.096)	3.678 (0.099)
SR-Gomea	5.998 (0.257)	4872 (252)	0.478 (0.011)	3.273 (0.078)	1.025 (0.052)	3.283 (0.071)
SR-Gomea-op	5.959 (0.143)	4625 (307)	0.477 (0.009)	**3.217 (0.072)**	0.991 (0.058)	3.287 (0.091)
BGLM	6.02 (0.47)	4872 (299)	0.484 (0.01)	3.547 (0.092)	0.986 (0.036)	3.287 (0.099)
SR-NP-trad	6.006 (0.473)	4842 (290)	0.489 (0.019)	3.536 (0.087)	0.975 (0.036)	3.312 (0.073)
SR-NP-max	6.043 (0.617)	4844 (318)	0.483 (0.011)	3.386 (0.063)	0.953 (0.069)	3.278 (0.063)
	Breast tumor	*Music*	*House*	*Wine*	*Toxicity*	*Gas*
OLS	9.8 (0.123)	1.028 (0.458)	35162 (1651)	0.736 (0.017)	1.264 (0.109)	7.169 (0.281)
GAM-splines	9.62 (0.067)	0.827 (0.031)	32633 (1111)	0.729 (0.019)	1.293 (0.168)	5.671 (0.168)
EBM	**9.493 (0.057)**	0.613 (0.058)	**31298 (944)**	**0.683 (0.019)**	**1.207 (0.114)**	**5.423 (0.119)**
SR-GP	10.242 (0.286)	0.762 (0.076)	49985 (7436)	0.807 (0.021)	1.395 (0.27)	9.966 (1.797)
SR-Gomea	9.798 (0.098)	0.624 (0.079)	33156 (863)	0.729 (0.017)	1.238 (0.066)	7.614 (0.205)
SR-Gomea-op	9.798 (0.103)	0.584 (0.06)	33091 (943)	0.734 (0.016)	1.273 (0.117)	7.747 (0.338)
B-GLM	9.798 (0.064)	0.721 (0.065)	40835 (1380)	0.739 (0.021)	1.293 (0.129)	6.47 (0.152)
SR-NP-trad	9.865 (0.141)	**0.550 (0.038)**	33922 (1158)	0.73 (0.019)	1.326 (0.125)	7.549 (0.515)
SR-NP-max	9.8 (0.117)	0.564 (0.059)	34021 (1251)	0.724 (0.018)	1.347 (0.199)	7.296 (0.728)

Table 6: Results obtained by cluster-specific interpretable models

	RMSE (std)					
	French Highway	*Insurance*	*Airbnb*	*Puma*	*Satellite*	*Wind*
SR-NP-naive-trad	6.446 (0.42)	6751 (521)	0.552 (0.017)	4.338 (0.229)	1.025 (0.04)	3.675 (0.139)
SR-NP-naive-max	6.377 (0.454)	6550 (365)	0.554 (0.036)	4.266 (0.18)	1.035 (0.148)	3.608 (0.079)
HSR-naive-trad	6.472 (0.462)	6429 (307)	0.584 (0.067)	4.156 (0.184)	1.012 (0.064)	3.547 (0.041)
HSR-naive-max	6.386 (0.473)	6328 (307)	0.545 (0.018)	4.062 (0.078)	0.991 (0.034)	3.562 (0.068)
SR-NP-trad	6.006 (0.473)	4842 (290)	0.489 (0.019)	3.536 (0.087)	0.975 (0.036)	3.312 (0.073)
SR-NP-max	6.043 (0.617)	4844 (318)	0.483 (0.011)	3.386 (0.063)	0.953 (0.069)	3.278 (0.063)
HSR-trad	5.921 (0.54)	**4840 (308)**	0.475 (0.012)	3.30 (0.084)	0.95 (0.034)	3.205 (0.074)
HSR-max	**5.80 (0.507)**	4844 (304)	**0.470 (0.011)**	**3.277 (0.082)**	**0.934 (0.056)**	**3.198 (0.074)**
	Breast tumor	*Music*	*House*	*Wine*	*Toxicity*	*Gas*
SR-NP-naive-trad	12.0 (0.161)	0.549 (0.05)	38810 (1892)	0.734 (0.027)	1.345 (0.173)	7.873 (1.55)
SR-NP-naive-max	11.953 (0.149)	0.576 (0.102)	38260 (1531)	0.723 (0.016)	1.262 (0.096)	7.105 (0.906)
HSR-naive-trad	11.932 (0.16)	0.524 (0.045)	37518 (1563)	0.735 (0.021)	1.363 (0.112)	7.46 (0.708)
HSR-naive-max	11.915 (0.108)	0.507 (0.056)	37640 (1741)	0.725 (0.019)	1.266 (0.071)	6.621 (0.15)
SR-NP-trad	9.865 (0.141)	0.550 (0.038)	33922 (1158)	0.73 (0.019)	1.326 (0.125)	7.549 (0.515)
SR-NP-max	9.8 (0.117)	0.564 (0.059)	34021 (1251)	0.724 (0.018)	1.347 (0.199)	7.296 (0.728)
HSR-trad	9.727 (0.056)	0.497 (0.05)	33542 (1127)	0.724 (0.019)	1.243 (0.097)	7.181 (0.618)
HSR-max	**9.662 (0.061)**	**0.471 (0.053)**	**33102 (833)**	**0.712 (0.014)**	**1.214 (0.065)**	**6.421 (0.150)**

Table 7: Comparison of two clustering strategies for the no pooling and partial pooling approaches: a naive one based on the original features versus the one based on the SHAP features (see Section 3.2)

4.8. Discussion

Confirming previous studies (Lou et al., 2012; Caruana et al., 2015), we observe that EBM, as a variant of GAM, is very efficient on all datasets. Moreover, this model meets many of the expected criteria for interpretability enumerated in (Arrieta et al., 2020). However, it also has limitations that can make it unsuitable for safety-critical systems. First, different optimization strategies adopted to learn an EBM model, can lead to different interpretations of its predictions (Chang et al., 2021). However, for safety-critical systems, trust in the identification of the main risk factors is required by experts when they elaborate remedial actions. Moreover, for satisfactory interpretability, it helps if a GAM has a small number of components and if each component function is relatively smooth. However, EBM, due to their reliance on boosted trees, can hardly maintain these constraints (Rudin et al., 2022). With our approach, field experts are more likely to be confident in models with cluster-specific behaviors and stable functional forms that highlight a selection of relevant factors and their interactions.

Furthermore, for the *French Highway* dataset, as already observed in (Veran et al., 2020), the best known strategy to estimate the number of crash counts is to average, for each highway network segment, the number of accidents that occurred in previous years (c.f., the *Local* model in Table 5). However, such a model does not offer much insight about the associations between crash counts and risk factors. We observed that flexible models, such as EBM, are able to approach in performance the local model by discovering quasi-identifiers of road segments. For example, an EBM discovers a complex nonlinear relationship between the altitude and the number of accidents, see Fig. 7. Accidents appear more likely for the lowest altitudes. However, this phenomenon should not be interpreted as a potential risk factor linked to the altitude. In fact, the model is using the altitude as a proxy variable to identify a group of nearby road segments. Therefore, in that particular context, EBM, despite its good predictive performance, does not always provide relevant information to field experts. It can even, at times, mislead them.

5. Dynamic interpretative process

5.1. Introduction

Although plots like the one of Fig. 7 make it possible to identify potentially misleading models' behaviors, the EBM model does not provide alternative associations between the explanatory variables and the target. With our approach, the risk of misinterpretation is reduced thanks to the successive models on the Pareto front: from less complex, which capture only overall effects, to most complex, which are flexible enough to focus on hazardous configurations specific to a few roadway segments. Through such a dynamic interpretative process, field experts can use the

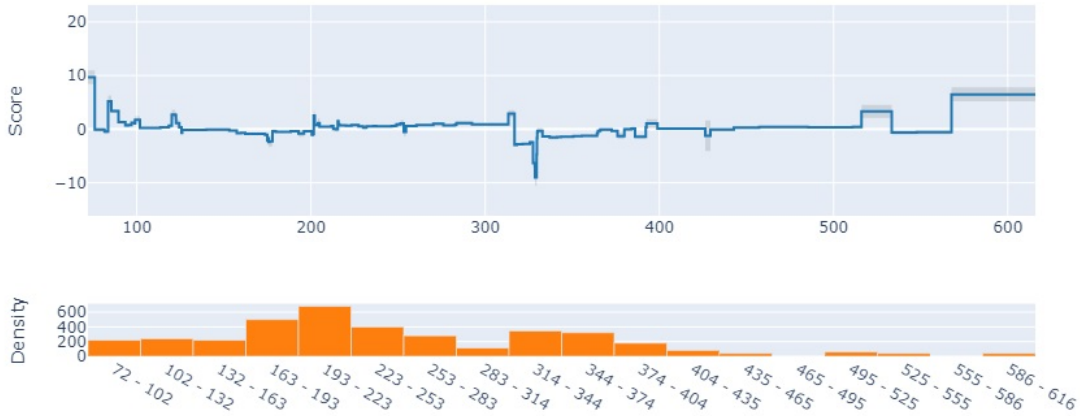


Figure 7: Global explanation plot provided by the InterpretML framework for an *EBM* model on the altitude variable on the *French Highway* dataset

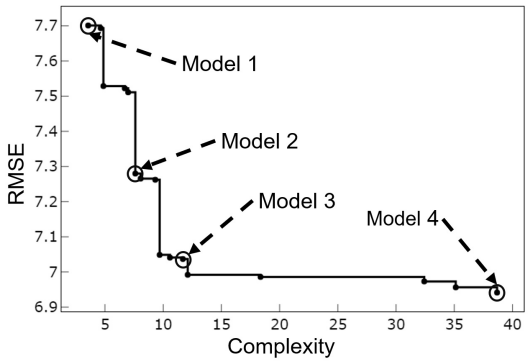
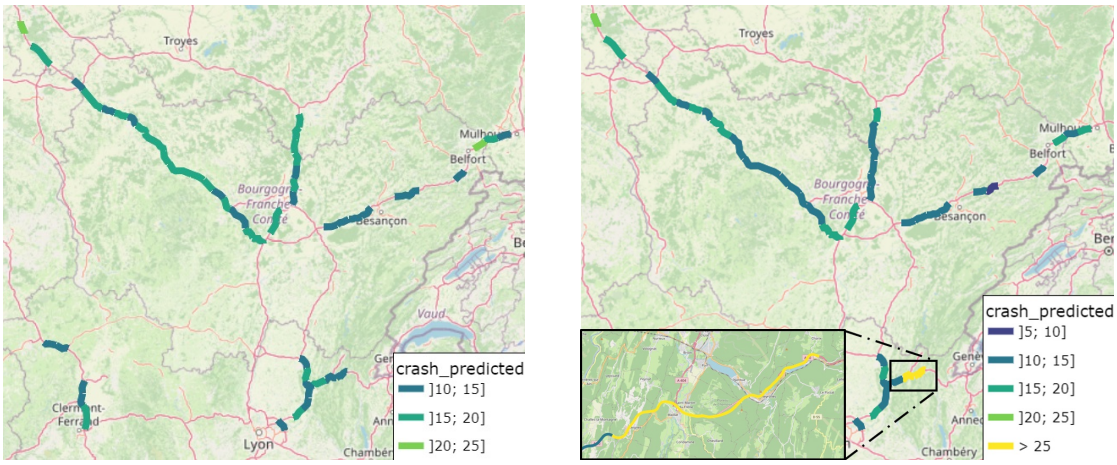


Figure 8: Pareto front for the selected cluster.

model best suited to meet their needs. In the next section, we illustrate this process on the *French Highway* dataset, supported by a graphical user interface developed in dialogue with field experts (see Appendix B). We suppose that our framework has already been trained on ten years of data, from January 1st, 2008 to December 31th, 2017. Data from 2018 is used to validate that, based on out-of-sample predictions, the framework provides useful information to safety experts. The first module of our framework, described in Section 3.2, identified four relevant clusters. For illustrative purposes, we focus on a moderately hazardous cluster, composed mainly of rural and mountainous segments.

5.2. From global to cluster-specific effects: an illustrative example

Safety experts can navigate within the series of cluster-specific models that make up the Pareto front (see Fig. 8), from the least complex one (viz. model 1) to the



(a) Model 1

(b) Model 2

Figure 9: Crash count predictions for 2018 on roadway segments belonging to the selected cluster.

most complex one (viz. model 4). Model 1 corresponds to the functional form of the global model (Section 3.4) whose coefficients are inferred based on cluster 0 data.

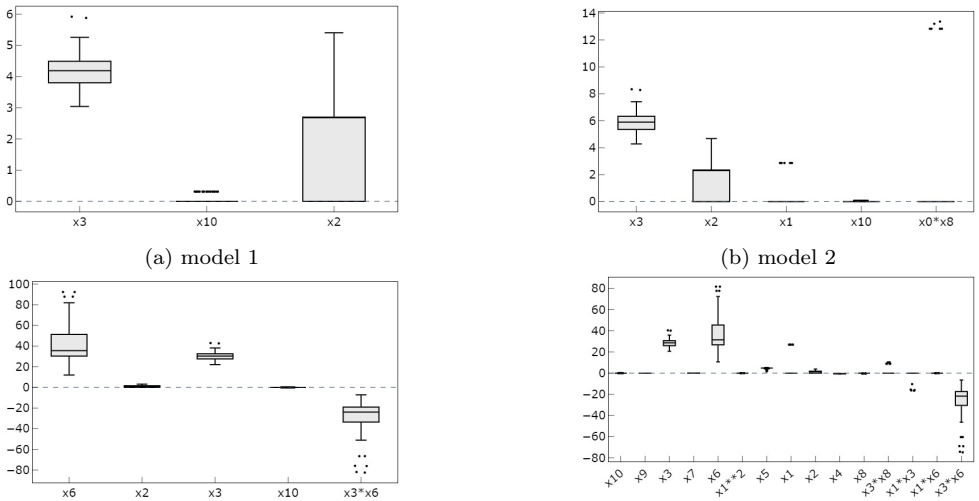
$$\text{model 1: } \hat{y} = 5.95 + 0.000394x_3 + 0.312x_{10} + 2.7x_2$$

with $\hat{y}$ being the predicted crash count, x_3 the average annual daily traffic, x_{10} the presence of bridges (binary) and x_2 the number of rest areas. From the effects plots of Fig. 10a, it appears that, for this model, the amount of traffic and the number of rest areas have the more prominent marginal effects.

Model 1 captures only the global risk factors. When considering models of increasing complexity, more specific effects will appear. For instance, model 2 (see Fig. 8) is defined as:

$$\text{model 2: } \hat{y} = 3.34 + 0.000555x_3 + 2.34x_2 + 1.43x_1 + 0.055x_{10} + 0.117x_0x_8$$

where the additional variables x_0, x_1 and x_8 are, respectively, the speed limit, the number of interchanges and the presence of tunnels. Out-of-sample predictions from models 1 and 2 differ locally (see Fig 9a and Fig 9b). In particular, in mountainous areas, segments considered as moderately hazardous by the first model, are now associated with a high risk of accidents due to the discovery of a first-order interaction between the presence of tunnel and the speed limit. Safety experts, by combining prior knowledge of the network with the observed transition from model 1 to model 2, are confident that this interaction is one of the main reasons why a large number of accidents have occurred on these segments during the ten years covered by the training data. This discovery may support a proposal for reducing the authorized speed limit on these specific segments.



(a) model 1 (b) model 2

(c) model 3 (d) model 4

Figure 10: Effects plots for the four selected models. Effects plots are obtained by computing for all observations the effect of a variable j on the crash count, defined by $\text{effect}_j^{(i)} = \beta_j x_j^{(i)}$, where β_j is the coefficient estimate of the j-th variable of the model and $x_j^{(i)}$ is the value of variable j on the i-th observation (Molnar, 2020).

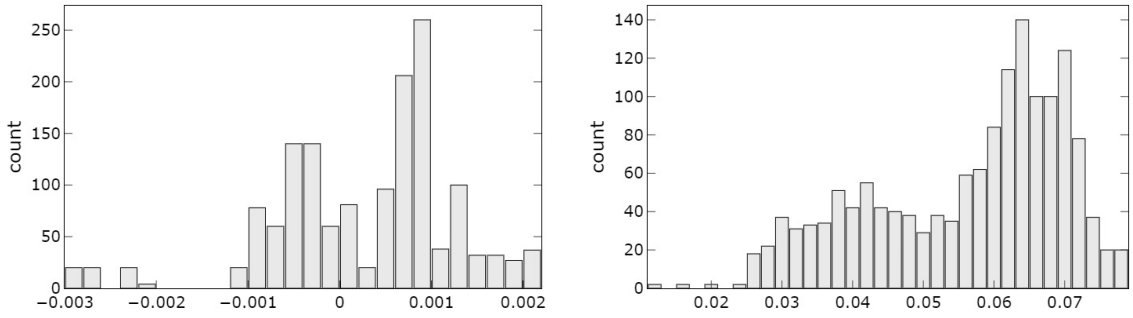
Nonetheless, one of the major difficulties for the interpretation of more complex models is related to the introduction of collinearities and interactions between continuous variables. To illustrate this, consider model 3 from Fig. 8:

$$\text{Model 3: } \hat{y} = -33.3 - 0.00489x_{10} + 1.57x_2 - 9.37 \cdot 10^{-6}x_3x_6 + 0.00286x_3 + 0.15x_6$$

Model 3 is characterized by an interaction between the averaged altitude x_6 and the traffic x_3. By focusing on this interaction, model 3 better fits training data than model 2 but its effects plots (see Fig 10c) are arguably more difficult to interpret. However, our framework only produces differentiable closed-form expressions for which it is always possible to compute the partial analytical derivatives (PD) w.r.t. variables of interest, to quantify explicitly their partial effects (i.e., a measure of the conditional effect of a variable on the target) (Aldeia and de França, 2021). In this sense, we can understand how a unit change in an explanatory variable affects the crash count when other variables are held constant. For instance, the partial derivatives for the trafic x_3 and altitude x_6 are respectively:

$$\text{PD}(x_3) = \frac{\delta\hat{y}}{\delta x_3} = 0.00286 - 9.37 \cdot 10^{-6}x_6, \quad \text{PD}(x_6) = \frac{\delta\hat{y}}{\delta x_6} = 0.15 - 9.37 \cdot 10^{-6}x_3$$

Histograms of the pointwise partial derivatives can be useful interpretative tools (see Fig. 11). Although the partial effects of the traffic x_3 are mostly positive, a few are



(a) PD(x_3) (b) PD(x_6)

Figure 11: Histograms of partial derivatives for (a) the traffic x_3 and (b) the altitude x_6, for model 3.

	x_3		x_6	
	overall	PD(x_3) < -0.0015	overall	PD(x_3) < -0.0015
count	1491	64	1491	64
mean	10001	12206	274	579
std	1465	997	105	32
min	7685	10877	79	521
max	14658	14658	616	616

Table 8: Description of explanatory variables for the overall cluster-specific data and for samples where partial derivatives w.r.t. x_3 are the lowest.

negative for segments of high altitudes and above average traffic (see Table 8). Thus, these variations in partial derivatives emphasize that the relation between the crash count and x_3 is more complex than the linear dependency proposed by model 1 and model 2. By introducing this novel interaction, model 3 manages to capture more variability in the dependent variable than the previous models.

Finally, model 4 is much more complex:

$$\begin{aligned}\text{Model 4: } \hat{y} = & -34.7 + 8.5 \cdot 10^{-6} x_1^2 - 0.000671 x_1 x_3 - 8.5 \cdot 10^{-6} x_1 x_6 + 13.5 x_1 \\ & + 0.0113 x_{10} + 1.92 x_2 - 8.5 \cdot 10^{-6} x_3 x_6 + 0.000679 x_3 x_8 + 0.00269 x_3 \\ & - 0.0276 x_4 + 1.61 x_5 + 0.133 x_6 + 0.12 x_7 - 0.276 x_8 + 0.0472 x_9\end{aligned}$$

As we can see from Fig. 10d, the introduction of new correlated terms improves slightly the fit. To capture extra variability in the dependent variable, model 4 introduces highly correlated combinations of terms. From model 3 to model 4, a sharp increase in complexity for a small gain in performance should alert the user to the risk of no longer understanding the inner workings of model 4: time must be spent at studying the various partial effects before deciding if the model can still be trusted.

5.3. Specificities and benefits of the ranking-by-complexity approach

Thanks to our complexity metric (see Eq. 2), model 3 does not dominate model 2 even though they have the same number of terms and their terms have equal complexities. If we had not penalize collinearities, then model 2, which is of high interest to field experts, would not have been included in the Pareto front. In this sense, the ranking-by-complexity favors a progressive analysis of numerous instructive models.

Among Fig. 8 models, some can attain similar predictive performances while bringing out different effects of the explanatory variables. This can be understood from the point of view of the Rashomon effect (Breiman, 2001) which characterizes problems where many accurate-but-different models exist to describe the same data (Semenova et al., 2019). As discussed by (Rudin, 2019), we argue that the availability of multiple efficient predictive models is useful since field experts may have more flexibility in choosing a model that they find interpretable. Moreover, we help them in this process as our definition of the complexity warns them when models are likely to be difficult to understand.

Finally, the dynamic interpretative process can be a useful tool to construct new handmade predictive models, based on the knowledge learnt by analyzing the Pareto optimal models. For instance, we have seen that the interactions introduced in model 2 and model 3 are both valuable. The user could consider building a new model with both of them.

5.4. Towards causality

A central question remains: among these different models, how can be distinguished the trustworthy ones from the ones based on spurious associations due to inductive bias? As illustrated above, one way is to rely on the diligence of the user equipped with expert knowledge and effective tools. This could also be partially automated when prior knowledge of the conditional independences between variables is formalized, e.g., as a causal graph (Pearl, 2009). Such approaches are beyond the scope of our current work. However, our framework fosters a dynamic interpretative process that, combined with a clear quantification of uncertainty, is a useful tool to identify variables of interest and to understand how they interact. Therefore, we can surmise that our framework could facilitate the development of causal models.

6. Conclusion

Predictive models are being used increasingly to make high stake decisions. For many applications, there is a need for both accuracy and interpretability. For instance, in highway safety analysis, we argue that a preference should be given to predictive models that are both accurate and fully interpretable in order to increase the confidence of safety experts in the identification of hazardous segments. Motivated by

these requirements, we propose an interpretable symbolic regression framework that first discovers a hierarchical structure in the data, and then learns global and cluster-specific models by means of a multi-objective simulated-annealing-based symbolic regression. More specifically, we first train a state-of-the-art non-parametric machine learning model and then compute for each observation the Shapley values of the explanatory variables. Based on the similarity induced by these Shapley values, we use an agglomerative clustering algorithm to partition the dataset. Moreover, through an original multi-objective symbolic regression, we compute a Pareto front of global predictive models. We select among these models the one offering a good trade-off between its predictive performance and its complexity. Afterwards, for each cluster of the previously discovered hierarchical structure, the global model is used as the starting seed for a new multi-objective symbolic regression. Finally, the best models, i.e. the ones appearing on the Pareto fronts, are re-estimated through Bayesian inference in order to associate uncertainty estimates to their coefficients.

On twelve regression datasets, the framework outperforms most interpretable models. On some datasets, we achieve performance comparable to that of non-parametric black box models. Furthermore, we presented a case study based on the highway network dataset to validate the new dynamic interpretative process made possible by our framework. As our approach discovers transparent and parsimonious symbolic models, safety experts can be more confident in their understanding of the relations between the explanatory variables and the dependent variable. Moreover, thanks to Bayesian inference, the risk factors are associated with measures of uncertainty. In addition, the use of Pareto optimization allows field experts to build a multi-scale view of the risk factors, from the most general to the most specific.

Our framework relies on a specific approach to discover a hierarchical structure. Even though we validated its robustness on numerous datasets, promising next steps involve analyzing other methods that are compatible with ours. For instance, in (Rengasamy et al., 2021b,a), the authors propose an efficient ensemble feature importance method where multiple feature importance approaches are applied to a set of ML models and their crisp importance values are combined to produce a final importance for each feature. Thus, we will constitute a benchmark of feature importance methods (Arrieta et al., 2020) and evaluate them based on their efficiency, scalability, and on the quality of computed clusters.

Finally, future work will extend the framework for near real-time crash risk assessment. In this context, since remedial actions will probably affect humans' lives even more directly, having both efficient and interpretable models will be all the more important to assist safety experts in their work.

Credit authorship contribution statement

Thomas Veran: Methodology, Writing – Original draft, Formal analysis, Software, Validation, Investigation, Visualization. **Pierre-Edouard Portier**: Conceptualization, Methodology, Investigation, Writing – Review & Editing, Supervision. **François Fouquet**: Methodology, Investigation, 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.

Acknowledgments

This work was supported by the French Association for Research and Technology (ANRT). We would like to thank APRR group, for giving us access to their data and for sharing with us their expertise in road safety.

References

Airbnb, 2019. New York city housing prices. URL: `https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data`. Last accessed: 10 March 2022.

Alaa, A.M., van der Schaar, M., 2019. Demystifying black-box models with symbolic metamodels. Advances in Neural Information Processing Systems 32, 11304–11314.

Aldeia, G.S.I., de França, F.O., 2021. Measuring feature importance of symbolic regression models using partial effects, in: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 750–758.

Arrieta, A.B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al., 2020. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. Information Fusion 58, 82–115.

Augusto, D.A., Barbosa, H.J., 2000. Symbolic regression via genetic programming, in: Proceedings. Vol. 1. Sixth Brazilian Symposium on Neural Networks, IEEE. pp. 173–178.

Borisov, V., Leemann, T., Seßler, K., Haug, J., Pawelczyk, M., Kasneci, G., 2021. Deep neural networks and tabular data: A survey. arXiv preprint arXiv:2110.01889 .

Breiman, L., 2001. Statistical modeling: The two cultures (with comments and a rejoinder by the author). Statistical science 16, 199–231.

Burlacu, B., Kronberger, G., Kommenda, M., 2020. Operon c++ an efficient genetic programming framework for symbolic regression, in: Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion, pp. 1562–1570.

Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., Elhadad, N., 2015. 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, pp. 1721–1730.

Chang, C.H., Tan, S., Lengerich, B., Goldenberg, A., Caruana, R., 2021. How interpretable and trustworthy are gams?, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 95–105.

Chang, L.Y., 2005. Analysis of freeway accident frequencies: negative binomial regression versus artificial neural network. Safety science 43, 541–557.

Chen, S.H., 2012. Genetic algorithms and genetic programming in computational finance. Springer Science & Business Media.

Chen, T., Guestrin, C., 2016. Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pp. 785–794.

Delahaye, D., Chaimatanan, S., Mongeau, M., 2019. Simulated annealing: From basics to applications, in: Handbook of metaheuristics. Springer, pp. 1–35.

El-Basyouny, K., Sayed, T., 2006. Comparison of two negative binomial regression techniques in developing accident prediction models. Transportation Research Record 1950, 9–16.

Eren, Y., İbrahim B. Küçükdemiral, İlker Üstoğlu, 2017. Chapter 2 - introduction to optimization, in: Erdinç, O. (Ed.), Optimization in Renewable Energy Systems. Butterworth-Heinemann, pp. 27–74.

Firinguetti, L., Bobadilla, G., 2011. Asymptotic confidence intervals in ridge regression based on the edgeworth expansion. Statistical Papers 52, 287–307.

Fletcher, R., 2013. Practical methods of optimization. John Wiley & Sons.

Friedman, J.H., 2001. Greedy function approximation: a gradient boosting machine. Annals of statistics , 1189–1232.

Gelman, A., Hill, J., 2006. Data analysis using regression and multilevel/hierarchical models. Cambridge university press.

Golub, G.H., Heath, M., Wahba, G., 1979. Generalized cross-validation as a method for choosing a good ridge parameter. Technometrics 21, 215–223.

Haeri, M.A., Ebadzadeh, M.M., Folino, G., 2017. Statistical genetic programming for symbolic regression. Applied Soft Computing 60, 447–469.

Hastie, T.J., Tibshirani, R.J., 2017. Generalized additive models. Routledge.

Hoffman, M.D., Gelman, A., et al., 2014. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. J. Mach. Learn. Res. 15, 1593–1623.

Huang, H., Abdel-Aty, M., 2010. Multilevel data and bayesian analysis in traffic safety. Accident Analysis & Prevention 42, 1556–1565.

Jin, Y., Fu, W., Kang, J., Guo, J., Guo, J., 2019. Bayesian symbolic regression. arXiv preprint arXiv:1910.08892 .

Jones, A.P., Jørgensen, S.H., 2003. The use of multilevel models for the prediction of road accident outcomes. Accident Analysis & Prevention 35, 59–69.

Jones, B., Janssen, L., Mannering, F., 1991. Analysis of the frequency and duration of freeway accidents in seattle. Accident Analysis & Prevention 23, 239–255.

Joshua, S.C., Garber, N.J., 1990. Estimating truck accident rate and involvements using linear and poisson regression models. Transportation planning and Technology 15, 41–58.

Keijzer, M., 2003. Improving symbolic regression with interval arithmetic and linear scaling, in: European Conference on Genetic Programming, Springer. pp. 70–82.

Khoda Bakhshi, A., Ahmed, M.M., 2021. Utilizing black-box visualization tools to interpret non-parametric real-time risk assessment models. Transportmetrica A: Transport Science 17, 739–765.

Kim, D.G., Lee, Y., Washington, S., Choi, K., 2007. Modeling crash outcome probabilities at rural intersections: Application of hierarchical binomial logistic models. Accident Analysis & Prevention 39, 125–134.

La Cava, W., Danai, K., Spector, L., Fleming, P., Wright, A., Lackner, M., 2016. Automatic identification of wind turbine models using evolutionary multiobjective optimization. Renewable Energy 87, 892–902.

La Cava, W., Orzechowski, P., Burlacu, B., de França, F.O., Virgolin, M., Jin, Y., Kommenda, M., Moore, J.H., 2021. Contemporary symbolic regression methods and their relative performance. arXiv preprint arXiv:2107.14351 .

Lakkaraju, H., Kamar, E., Caruana, R., Leskovec, J., 2019. Faithful and customizable explanations of black box models, in: Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, pp. 131–138.

Lantz, B., 2013. Medical cost personal dataset. URL: `https://github.com/stedy/Machine-Learning-with-R-datasets`. Last accessed: 10 March 2022.

Li, X., Lord, D., Zhang, Y., Xie, Y., 2008. Predicting motor vehicle crashes using support vector machine models. Accident Analysis & Prevention 40, 1611–1618.

Lipton, Z.C., 2018. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. Queue 16, 31–57.

Lord, D., Kuo, P.F., 2012. Examining the effects of site selection criteria for evaluating the effectiveness of traffic safety countermeasures. Accident Analysis & Prevention 47, 52–63.

Lord, D., Manar, A., Vizioli, A., 2005. Modeling crash-flow-density and crash-flow-v/c ratio relationships for rural and urban freeway segments. Accident Analysis & Prevention 37, 185–199.

Lord, D., Mannering, F., 2010. The statistical analysis of crash-frequency data: A review and assessment of methodological alternatives. Transportation research part A: policy and practice 44, 291–305.

Lou, Y., Caruana, R., Gehrke, J., 2012. Intelligible models for classification and regression, in: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 150–158.

Lundberg, S.M., Erion, G., Chen, H., DeGrave, A., Prutkin, J.M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., Lee, S.I., 2020. From local explanations to global understanding with explainable ai for trees. Nature machine intelligence 2, 56–67.

Lundberg, S.M., Erion, G.G., Lee, S.I., 2018. Consistent individualized feature attribution for tree ensembles. arXiv preprint arXiv:1802.03888 .

Lundberg, S.M., Lee, S.I., 2017. A unified approach to interpreting model predictions, in: Proceedings of the 31st international conference on neural information processing systems, pp. 4768–4777.

McCullagh, P., Nelder, J.A., 2019. Generalized linear models. Routledge.

McCullough, B., Vinod, H., 1998. Implementing the double bootstrap. Computational Economics 12, 79–95.

McKay, B., Willis, M.J., Barton, G.W., 1995. Using a tree structured genetic algorithm to perform symbolic regression, in: First international conference on genetic algorithms in engineering systems: innovations and applications, IET. pp. 487–492.

Mehta, P., Bukov, M., Wang, C.H., Day, A.G., Richardson, C., Fisher, C.K., Schwab, D.J., 2019. A high-bias, low-variance introduction to machine learning for physicists. Physics reports 810, 1–124.

Meier, A., Gonter, M., Kruse, R., 2014. Symbolic regression for precrash accident severity prediction, in: International Conference on Hybrid Artificial Intelligence Systems, Springer. pp. 133–144.

Meurer, A., Smith, C.P., Paprocki, M., Čertík, O., Kirpichev, S.B., Rocklin, M., Kumar, A., Ivanov, S., Moore, J.K., Singh, S., et al., 2017. Sympy: symbolic computing in python. PeerJ Computer Science 3, e103.

Miaou, S.P., 1994. The relationship between truck accidents and geometric design of road sections: Poisson versus negative binomial regressions. Accident Analysis & Prevention 26, 471–482.

Miaou, S.P., Lord, D., 2003. Modeling traffic crash-flow relationships for intersections: dispersion parameter, functional form, and bayes versus empirical bayes methods. Transportation Research Record 1840, 31–40.

Mihaita, A.S., Liu, Z., Cai, C., Rizoiu, M.A., 2019. Arterial incident duration prediction using a bi-level framework of extreme gradient-tree boosting. arXiv preprint arXiv:1905.12254 .

Molnar, C., 2020. Interpretable machine learning. Lulu. com.

Nori, H., Jenkins, S., Koch, P., Caruana, R., 2019. Interpretml: A unified framework for machine learning interpretability. arXiv preprint arXiv:1909.09223 .

Parsa, A.B., Movahedi, A., Taghipour, H., Derrible, S., Mohammadian, A.K., 2020. Toward safer highways, application of xgboost and shap for real-time accident detection and feature analysis. Accident Analysis & Prevention 136, 105405.

Patelli, A., Lush, V., Ekart, A., Ilie-Zudor, E., 2020. Traffic modelling and prediction via symbolic regression on road sensor data. arXiv preprint arXiv:2002.06095 .

Pearl, J., 2009. Causality. Cambridge university press.

Petersen, B.K., Larma, M.L., Mundhenk, T.N., Santiago, C.P., Kim, S.K., Kim, J.T., 2019. Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients. arXiv preprint arXiv:1912.04871 .

Poursabzi-Sangdeh, F., Goldstein, D.G., Hofman, J.M., Wortman Vaughan, J.W., Wallach, H., 2021. Manipulating and measuring model interpretability, in: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, pp. 1–52.

Rengasamy, D., Mase, J.M., Torres, M.T., Rothwell, B., Winkler, D.A., Figueredo, G.P., 2021a. Mechanistic interpretation of machine learning inference: A fuzzy feature importance fusion approach. arXiv preprint arXiv:2110.11713 .

Rengasamy, D., Rothwell, B.C., Figueredo, G.P., 2021b. Towards a more reliable interpretation of machine learning outputs for safety-critical systems using feature importance fusion. Applied Sciences 11, 11854.

Ribeiro, M.T., Singh, S., Guestrin, C., 2016. " 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, pp. 1135–1144.

Rudin, C., 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence 1, 206–215.

Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., Zhong, C., 2022. Interpretable machine learning: Fundamental principles and 10 grand challenges. Statistics Surveys 16, 1–85.

Schmidt, M., Lipson, H., 2009. Distilling free-form natural laws from experimental data. science 324, 81–85.

Semenova, L., Rudin, C., Parr, R., 2019. A study in rashomon curves and volumes: A new perspective on generalization and model simplicity in machine learning. arXiv preprint arXiv:1908.01755 .

Smith, K.I., Everson, R.M., Fieldsend, J.E., 2004. Dominance measures for multi-objective simulated annealing, in: Proceedings of the 2004 congress on evolutionary computation (IEEE Cat. No. 04TH8753), IEEE. pp. 23–30.

Sobol, I.M., 2001. Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. Mathematics and computers in simulation 55, 271–280.

Stanislawska, K., Krawiec, K., Kundzewicz, Z.W., 2012. Modeling global temperature changes with genetic programming. Computers & Mathematics with Applications 64, 3717–3728.

Stinstra, E., Rennen, G., Teeuwen, G., 2008. Metamodeling by symbolic regression and pareto simulated annealing. Structural and Multidisciplinary Optimization 35, 315–326.

Tan, S., Caruana, R., Hooker, G., Koch, P., Gordo, A., 2018. Learning global additive explanations for neural nets using model distillation. arXiv preprint arXiv:1801.08640 .

Teney, D., Peyrard, M., Abbasnejad, E., 2022. Predicting is not understanding: Recognizing and addressing underspecification in machine learning. arXiv preprint arXiv:2207.02598 .

Thorndike, R.L., 1953. Who belongs in the family? Psychometrika 18, 267–276.

Tomsett, R., Preece, A., Braines, D., Cerutti, F., Chakraborty, S., Srivastava, M., Pearson, G., Kaplan, L., 2020. Rapid trust calibration through interpretable and uncertainty-aware ai. Patterns 1, 100049.

Udrescu, S.M., Tan, A., Feng, J., Neto, O., Wu, T., Tegmark, M., 2020. Ai feynman 2.0: Pareto-optimal symbolic regression exploiting graph modularity. arXiv preprint arXiv:2006.10782 .

Udrescu, S.M., Tegmark, M., 2020. Ai feynman: A physics-inspired method for symbolic regression. Science Advances 6, eaay2631.

Valipour, M., You, B., Panju, M., Ghodsi, A., 2021. Symbolicgpt: A generative transformer model for symbolic regression. arXiv preprint arXiv:2106.14131 .

Veran, T., Portier, P.E., Fouquet, F., 2020. Crash prediction for a french highway network with an xai-informed bayesian hierarchical model, in: 2020 IEEE International Conference on Big Data (Big Data), IEEE. pp. 1256–1265.

Vinod, H.D., 1995. Double bootstrap for shrinkage estimators. Journal of Econometrics 68, 287–302.

Virgolin, M., Alderliesten, T., Witteveen, C., Bosman, P.A., 2021. Improving model-based genetic programming for symbolic regression of small expressions. Evolutionary computation 29, 211–237.

Yu, R., Abdel-Aty, M., 2013. Utilizing support vector machine in real-time crash risk evaluation. Accident Analysis & Prevention 51, 252–259.

Zeng, Q., Huang, H., Pei, X., Wong, S., Gao, M., 2016. Rule extraction from an optimized neural network for traffic crash frequency modeling. Accident Analysis & Prevention 97, 87–95.

Appendix A. Hyper-parameters tuning

Model	Hyper-parameters	Values
GAM-splines	lam	$\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$
EBM	max_bins	$\{8, 16, 32, 64, 128, 256, 512, 1024\}$
	min_samples_leaf	$\{1, 2, 5, 10, 20\}$
SR-GP	population_size	$\{500, 1000, 1500\}$
	generations	$\{20, 50, 100\}$
SR-Gomea	initmaxtreeheight	$\{4, 6\}$
	popsize	$\{500, 1000\}$
XGBoost	learning_rate	$\{0.0001, 0.001, 0.01, 0.1\}$
	max_depth	$\{2, 3, 5, 10, 15\}$
	min_child_weight	$\{1, 3, 5, 7\}$
	gamma	$\{0, 0.5, 1, 1.5, 2, 5\}$
	col_sample_by_tree	$\{0.3, 0.4, 0.5, 0.7, 1\}$
SVM	C	$\{0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000\}$
	kernel	$\{\text{linear}, \text{poly}, \text{rbf}\}$
	tol	$\{0.0000001, 0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1\}$
MLP	hidden_layer_sizes	$\{(16, 16), (16, 8), (8, 8)\}$
	alpha	$\{0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1\}$
	activation	$\{\text{tanh}, \text{relu}\}$
	learning_rate_init	$\{0.0001, 0.001, 0.01, 0.1\}$

Appendix B. Data visualization tool