

Article

Towards Model-Driven Explainable Artificial Intelligence: Function Identification with Grammatical Evolution

Dominik Sepiolo *  and Antoni Ligeza AGH University of Krakow, Department of Applied Computer Science, al. Mickiewicza 30,
30-059 Krakow, Poland

* Correspondence: sepiolo@agh.edu.pl

Abstract: Machine learning is a well-matured discipline, and exploration of datasets can be performed in an efficient way, leading to accurate and operational prediction and decision models. On the other hand, most methods tend to produce black-box-type models, which can be considered a serious drawback. This is so, since in case of numerous practical applications, it is also required to justify, explain, and uncover the inner decision mechanism so that an in-depth understanding of the causal and functional dependencies becomes possible and some responsibility for the decision can be considered. This paper addresses the critical need for model-driven eXplainable Artificial Intelligence (XAI) by exploring the limitations inherent in existing explanatory mechanisms, such as LIME or SHAP, which rely solely on input data. This seems to be an intrinsic limitation and a conceptual error, as no expert domain knowledge can come into play, and no analytical models of the phenomena under investigation are created. In order to deal with this issue, this paper puts forward the idea of building open, white-box explanatory models. To do that, we propose employing grammatical evolution tools combined with expert domain knowledge. The results demonstrate that the developed models can effectively explain the structure and behavior of decision models in terms of components, connections, causality, and simple functional dependencies.

Keywords: machine learning; explainable artificial intelligence; model-driven explainable artificial intelligence; causal reasoning; functional dependencies; model discovery; grammatical evolution



Citation: Sepiolo, D.; Ligeza, A. Towards Model-Driven Explainable Artificial Intelligence: Function Identification with Grammatical Evolution. *Appl. Sci.* **2024**, *14*, 5950. <https://doi.org/10.3390/app14135950>

Academic Editor: Manuel Graña

Received: 29 March 2024

Revised: 31 May 2024

Accepted: 3 July 2024

Published: 8 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Machine Learning (ML) has become a well-matured sub-discipline of Artificial Intelligence (AI) with a large collection of problem-solving tools and an extensive range of practical applications [1]. Domains such as biomedical data analysis, natural language processing, and large-scale technological systems have reported numerous successful projects and implementations [2–5], along with significant progress regarding the variety of ML algorithms and tools. However, it appears that the traditional machine learning paradigm, e.g., as outlined in [6,7], which focuses on developing a finite-set decision-making classifier or value prediction model, remains somewhat too restricted and conservative. There has been little to no conceptual progress observed concerning the problem formulation [8] within this framework. ML data repositories are typically based on the same simple scheme of attributive decision tables, without incorporating any other knowledge-based components. This seems to constitute a serious and inherent drawback when considering the need to comprehend the semantics of the induced models and justify, explain and validate the produced predictions and decisions.

Especially when encountering some of the most challenging AI issues of today, which involve real understanding how intelligent systems function (e.g., model discovery for model-based reasoning [9], explainable artificial intelligence (XAI) [10,11], trustworthy decision-making, and interpretable and explainable AI (<https://www.bmc.com/blogs/machine-learning-interpretability-vs-explainability/> (accessed on 29 March 2024))),

among others), the advantages of some model-driven approaches seem to be hard to overestimate. To advance towards the building of model-driven XAI, incorporating knowledge-based components and more advanced function identification methods (e.g., symbolic regression and grammatical evolution) appears necessary and promising.

Explainable artificial intelligence (XAI) aims to deliver AI solutions, decisions, and predictions that are comprehensible to humans. Most current ML techniques, such as deep learning, rely on black-box models. To ensure an appropriate level of transparency, as well as justifiability and trustworthiness, it is crucial for humans to understand the underlying rules of the game.

XAI includes both transparent models and post hoc explainability techniques. Transparent models (e.g., decision trees) are inherently interpretable (explainable by design), allowing direct insight into their decision-making processes. In contrast, post hoc techniques are focused on generating explanations based on the model output rather than its internal architecture. For example, some common strategies include generating visual explanations and employing surrogate models.

Various taxonomies can be proposed for post hoc explainability techniques [10–16]. The most popular ones are listed as follows:

- Local (explaining individual prediction) vs. global explanations (providing an overview of the model of the decision-making process);
- Model-agnostic (can be applied to any model, regardless of its internal workings) vs. model-specific explanations (customized for distinct model architectures).

The current approaches to XAI often employ shallow models, which prioritize simplicity and interpretability over accuracy, constructing an easily comprehensible model on top of one derived from classical ML techniques. They operate on a subset of the original input data and lack the ability to integrate auxiliary knowledge. A broad and representative selection of such methods and tools is discussed in [10,11].

Local Interpretable Model-Agnostic Explanations (LIMEs) [17] exemplify the concept of explanation by simplification. The LIME algorithm creates an explanation for an individual prediction by creating a simple, interpretable linear model that approximates the behavior of the opaque model in the vicinity of the prediction. Another significant technique is SHapley Additive exPlanations (SHAPs) [18], which use a game-theoretic approach (Shapley values) to demonstrate feature importance for each prediction. Additional methods include visualization techniques and explanations by example. We provided a broad, critical overview of the current shallow approaches to XAI in [19,20].

The previously mentioned approaches are considered shallow because they offer explanations at the same abstraction level at which decisions are made, without involving a deeper theoretical investigation or understanding of the model's underlying principles and external declarative knowledge. An inherent limitation of shallow methods is their inability to fully capture the model's underlying complexity. Therefore, it is crucial that these methods be used in conjunction with deeper, more transparent techniques to achieve a more comprehensive understanding of the model's behavior and structure.

Genetic Programming (GP) is increasingly used within the field of XAI, predominantly through two primary methodologies that are in line with the current XAI taxonomy, namely intrinsic and post hoc interpretability. The intrinsic interpretability approach involves the development of GP models that are inherently interpretable by design. Such models leverage symbolic components that naturally facilitate human comprehension. On the other hand, post hoc interpretability in GP focuses on explaining (globally and locally) models typically considered opaque by simplifying or approximating their decision-making processes into forms that are more comprehensible to users, such as decision trees or linear models [21]. GP is generally considered to be explainable, particularly when compared to other ML methods (e.g., deep learning). However, more complex evolutionary models can be challenging to understand, and different XAI techniques can be applied in order to make them interpretable [22].

This paper explores the issue of model-driven explainable artificial intelligence. In fact, the proposed approach is related to concepts such as model discovery, model-based reasoning [9], symbolic regression, causal modeling [23], function discovery, and discovery by design [24]. As mentioned before, machine learning is a well-matured discipline, and exploration of even large datasets can be performed in an efficient way, leading to accurate and operational prediction and decision models. On the other hand, most methods tend to produce black-box-type models only. In the cases of numerous practical applications, it is also required to uncover the inner decision/prediction mechanism so that an in-depth understanding of the causal and functional dependencies becomes possible and some justification or responsibility for the decision can be considered. This is so in cases where justification, explanation, and validation of the produced results are required, e.g., for confirmation of correctness, safety, or reliability. Unfortunately, the explanatory mechanisms developed so far, such as LIME or SHAP, are based on the same input data that are the input for machine learning algorithms. This seems to be an in-born limitation and an intrinsic conceptual error, as no expert domain knowledge can come into play. In this paper, the idea of building open, white-box explanatory models is put forward. To do that, we propose employing grammatical evolution tools supported by expert domain knowledge. The resulting models are aimed at explaining the structure and behavior of the decision model in terms of components, connections, and simple functional dependencies. The primary contribution of this study is three-fold and summarized as follows:

- We clearly put forward both the need for model-driven explainable artificial intelligence (in opposition to existing post hoc approaches), as well as the conceptual framework underlying this technology;
- We outline the approach to put it into practice in the form of the 3C principle, which points to the need to define causality, components, and structure (connections) to build an explainable model of the analyzed phenomenon;
- Finally, we point to grammatical evolution as a rational tool of choice for identification of analytical models.

In fact, the proposed framework integrates causal relationships, functional dependencies, grammatical evolution, and a priori expert knowledge to enhance the interpretability and efficacy of AI models. While GE has been widely applied for feature construction tasks [21], we postulate that a knowledge-based component (ingested by grammar) should be part of the meaningful intermediate variable construction process.

This paper is structured as follows: Section 2 is subdivided into several parts, each detailing a component of the methodology, namely (i) an example network that introduces a diagnostic example of a feed-forward arithmetic circuit to illustrate the concept, (ii) a motivational example that presents data and preliminary analysis to motivate the study, and (iii) an explanation and application of grammatical evolution to develop a model with enhanced explainability. A conceptual framework for model-driven XAI and its application to the generation of explanations for the Iris dataset are presented in Section 3. Section 4 reports on the implications of using MD-XAI, highlighting the advantages of GE in creating interpretable models, as well as the encountered challenges and limitations. Section 5 summarizes the key findings and contributions of this study and suggests directions for future research.

2. Materials and Methods

In this section, we briefly explain some ideas behind the model-driven explanation concept. A classical example from the domain of model-based diagnosis serves as a core of the discussion. The main aim is to provide some intuitions with respect to how a transparent, explanatory model should look.

2.1. Example Network

Let us briefly recapitulate a classical example from the artificial intelligence domain applied to diagnostic reasoning. This example is based on analysis of the behavior of a

multiplier–adder circuit; it was presented and analyzed in a seminal paper by R. Reiter [25]. It has also been widely explored in a number of AI diagnostic readings, analyses, and investigations, such as [26], as well as some of the second author’s papers, including [27–29].

Here, we focus on its causal structure and functional components.

The system has 5 numerical inputs, namely A, B, C, D, and E, and two numerical outputs, namely F and G. As the very first approach, it can be modeled as a single black box of 5 inputs and 2 outputs, perhaps equipped with an input–output data table of arbitrary length specifying its behavior. However, such an approach would not be useful in understanding how the system works and how the output is produced.

A better and deeper model of the system can be specified with a Directed Acyclic Graph (DAG), as shown in Figure 1.

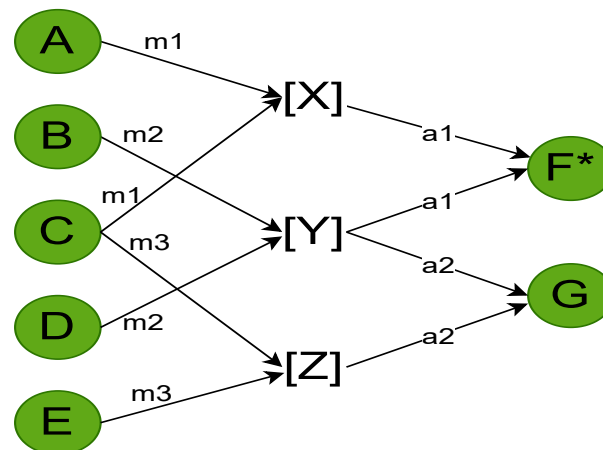


Figure 1. An example of a logical causal graph, namely a multiplier–adder system; a variable observed to misbehave is marked with an asterisk symbol (F^*).

The system has five input variables, namely A, B, C, D, and E. They influence further intermediate variables, namely X, Y, and Z (the internal variables). Finally, these variables shape the output variables, namely F and G.

A more detailed model of the system is composed of two functional layers. The first one contains three multipliers, namely $m1$, $m2$, and $m3$, receiving the input signals, namely A, B, C, D, and E. The second layer is composed of two adders, namely $a1$ and $a2$, producing the output values of F and G. Only inputs (of the first layer) and outputs (of the second layer) of the system are directly observable. The intermediate variables, namely X, Y, and Z, are hidden and normally cannot be observed.

A schema of the circuit, presenting its inputs, intermediate variables, outputs, components, and connections, is presented in Figure 2.

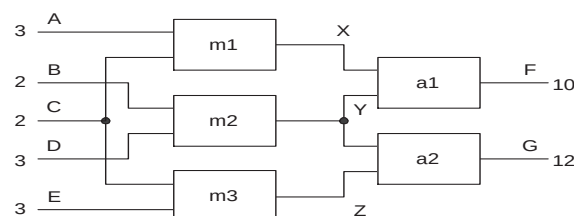


Figure 2. An example of a simple network of components, namely the multiplier–adder case.

This kind of model, containing functional blocks as components, as well as specification of intermediate variables and component connections, is the intuitive model we aim to use for almost any kind of explanatory reasoning. Such a model can be used for output prediction, abductive reasoning for finding possible inputs and generating the observed

output, and even for complex diagnostic reasoning [25,27]. Such a case is considered below. The aim is to show the potential of using such models.

Under normal conditions, the presented model allows for full explanation of the behavior of the system so that the simulation or prediction task can be performed and explained. This is enough to consider the current input values and propagate them forwards in a systematic way so that the values of the output are received. In our case, we have $A = 3$, $B = 2$, $C = 2$, $D = 3$, and $E = 3$; if all the components are working correctly, the predicted output values should be $F = 12$ and $G = 12$. But having such a detailed model, one can perform much more advanced diagnostic reasoning.

Since $F = 10$, we can conclude that the system is faulty. In model-based diagnosis, it is typically assumed that faulty behavior is caused by a fault of a named component or a simultaneous fault of a set of such components; no faults caused by faulty links, parameter settings, or the internal structure are considered. A simple analysis of the scheme presented in Figure 2 leads to the conclusion that one of the components producing the output at F ($m1$, $m2$, or $a1$) must be faulty. In fact, $C_1 = \{m1, m2, a1\}$ forms the so-called conflict set [25], i.e., a subset of system components such that at least one of them is sure to be faulty. Also note that only $m1$ and $a1$ being faulty can be considered a justified explanation for the observed fault. Assuming that $m2$ alone is faulty is not a justified explanation for faulty behavior, since it is not consistent with the observation of $G = 12$.

A bit more complicated analysis results in the observation that there exists another conflict set, namely $C_2 = \{m1, a1, a2, m3\}$ [25,27]; if $m3$ and $a2$, are correct, the observation that $G = 12$ leads to the conclusion that $Y = 6$, and this is in conflict with the assumption that $m1$ and $a1$ are correct, yielding $X = 6$ and $Y = 4$.

At this point, it is worth stating that C_1 and C_2 are of different types and origins. Here, C_1 can be classified as the causal type (all the components directly influence the value of F). On the other hand, C_2 is more of a constraint type, as it is discovered by the observation that some mathematical constraint induced by the structure of the circuit is violated by the current observations.

In the analyzed case, i.e., F being faulty and G being correct, the final (minimal) diagnoses fully explaining the observed misbehavior for the considered case are calculated as reduced elements of the Cartesian product of $C_1 = \{m1, m2, a1\}$ and $C_2 = \{m1, m3, a1, a2\}$ [25,27]; these are the so-called *hitting sets* [25]. The following potential diagnoses are considered: $D_1 = \{m1\}$, $D_2 = \{a1\}$, $D_3 = \{a2, m2\}$, and $D_4 = \{m2, m3\}$. All the diagnoses, although hypothetical, fully explain the observed misbehavior of the system.

The above example was presented in detail to introduce the motivation for a detailed system specification for explanatory reasoning. In fact, one needs to know the components, connections, and causal dependencies to fully understand the work of a complex system. Some interesting analyses of the system concerning identification and explanation of numerical faults are presented in [28,29], and an investigation of possible model discovery is presented in [30].

2.2. A Motivation Example

In this section, we present a comprehensive motivation example. The role of this example is to (i) introduce new ideas and some terminology underlying the proposed approach, (ii) show the way of dealing with the data and the necessity of supplementary knowledge introduction, and (iii) show why classical approaches to ML/XAI have some intrinsic limitations.

The data table below is used as input for ML algorithms.

In fact, Table 1 gives only an idea of how the data look and covers only a very small sample of the data in use. A complete data table consists of 100 observations.

Table 1. A schematic presentation of the input data. The table contains a sample of the data used for experiments.

a	b	c	Decision
3	9	7	0
1	4	4	1
5	9	4	2
⋮	⋮	⋮	⋮

The structure of the data includes three independent input columns (labeled *a*, *b*, and *c*) and one output or decision column (labeled decision). The input columns generally contain real numbers, whereas the output is an integer with possible values of 0, 1, and 2. Consequently, the input triples are deterministically classified into three distinct classes. The ML task is to create a prediction model that can determine the output value (decision) for any given set of input values.

Some ML paradigms selected for this experiment are Linear Regression (LR), Decision Trees (DT), and Random Forest (RF). A series of experiments implemented in the R language (using the stats and caret packages) was carried out in order to create models that attempt to discover the decision function from example data. A total of 100 observations were employed for training purposes, and the models were subsequently validated on an additional set of 100 observations, which were distributed similarly to the training dataset. Results of the experiments were assessed using the following metrics: Root Mean Squared Error (RMSE), calculated as $RMSE(y, \hat{y}) = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$, and RMSE normalized by the mean ($NRMSE = \frac{RMSE}{\bar{y}}$). The results are presented in Table 2 below.

Table 2. RMSE for selected ML techniques.

	RMSE Train	NRMSE Train	RMSE Test	NRMSE Test
Linear Regression	0.8267	0.8026	0.8239	0.8368
Decision Tree	0.5431	0.5273	0.8142	0.8224
Random Forest	0.2284	0.2218	0.5883	0.5942

The experiment shows that the prediction results are far from being perfect. None of the proposed techniques is capable of creating an accurate ML model. While LR returns the mean decision value of training data and performs similarly on the test dataset, the DT and RF models are overfit.

For the learned RF model, we applied the most popular XAI tools, namely LIME and SHAP. The results for representative prediction are presented in Figure 3 below.

LIME suggests variable *b* has the biggest impact on the prediction, while variable *c* contributes the least. In contrast to that, the SHAP explanation puts forward *c* as the most impactful variable and *b* as the least important. Moreover, in LIME, both *b* and *c* make positive contributions to the outcome of the model, while in SHAP, the contribution of both is negative. As there are considerable discrepancies between the explanations proposed by LIME and SHAP, it is not straightforward to decide which explanation should be respected as the correct one. Moreover, there is no single explanation technique that is universally superior, and there are no common choice criteria with respect to which technique should be used for explanation generation for the model. This is the reason a clear definition of the purpose of the XAI task and key properties of explanation for a given case is needed.

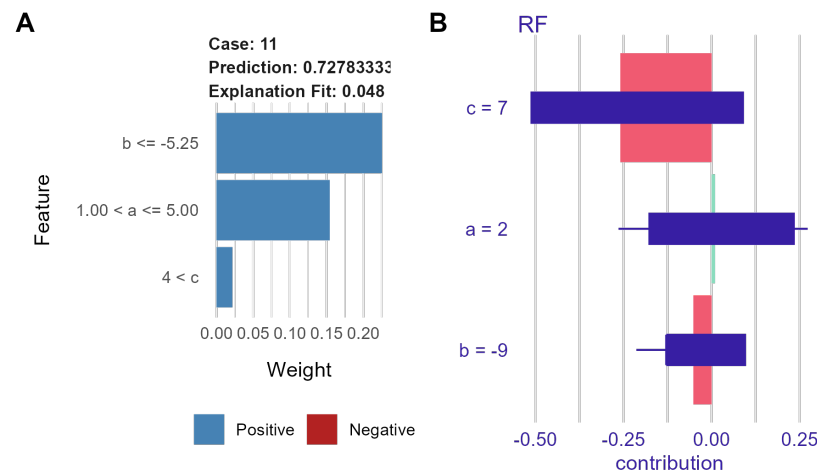


Figure 3. (A) LIME explanation and (B) SHAP explanation for a given prediction (green indicates positive contribution, while pink negative).

In conclusion, it becomes apparent that this simple example exposes the nonsense of trying to construct explanations without taking into account the semantics of the data.

Continuing this issue, note that while it is easy to measure the accuracy of the model, there is no consensus on the quality measures of an explanation. Different metrics, e.g., fidelity, consistency, simplicity, generality, reliability, and causality, have been developed in order to assess explanation quality [31,32]. If ML outputs align with intuitive human reasoning, users are more likely to trust and effectively interact with the system. A human-centric approach to explanation helps bridge the gap between complex ML algorithms and practical applications, ensuring that technology serves and aligns with human values and ethical standards. A model that performs well is generally considered more trustworthy than a model that performs poorly. However, the trustworthiness of a model goes far beyond its performance and depends on various factors. For this reason, it may be beneficial to explain less accurate models, as the explanation may help identify potential reasons for the poor performance and highlight areas that need improvement or may lead to the rejection of the model based on wrong assumptions. Nevertheless, the question of whether the explanation of a wrong model may be correct is still an open issue.

Now, let us redefine the experiment using the same input data table but according to the enhanced and extended methodology focused on in this paper, i.e., model-driven XAI (MD-XAI).

The core concept of the proposed MD-XAI approach involves integrating a knowledge component (KC). The KC comprises a set of definitions for (local) functional dependencies among the input variables (The notion of a functional dependency comes from the Relational databases). A variable (Y) is functionally dependent on variables $X_1, X_2, \dots, X_n$ if, given the values of $X = [X_1, X_2, \dots, X_n]$, the value of Y is uniquely determined. This relationship is represented as $X \rightarrow Y$. For obvious reasons, we only consider minimal functional dependencies, meaning that none of the X_i can be removed from X without losing the functional dependency.

Domain experts can provide some of the functional dependencies, which may be either established knowledge in the field or hypothetical statements (thus, requiring validation). Such hypothetical functional dependencies can also be found with help of specialized algorithms; a good review of such algorithms for causal structure discovery is provided in [33].

When discovering functional dependencies in the data, one can generally look for the following:

- Internal functional dependencies, i.e., among the variables naming the columns in the original set of input data;

- External functional dependencies, i.e., concerning the situation when a new, external variable is introduced, and it aggregates the knowledge represented by one or several original variables;
- Functional dependencies among the external variables, which can also be searched for, leading to a kind of multi-level scheme of dependencies [34].

Since variables a , b , and c are independent, no internal functional dependency exists among them (according to the knowledge provided by a domain expert). However, using expert knowledge, one can suggest introducing an external knowledge aggregation variable (d) such that $[a, b, c] \rightarrow d$. The value of d is determined by the following formula defined by an expert: $d = b^2 - 4ac$. This slightly enigmatic example should now become clear.

The initial example is rearranged by incorporating the KC. First, the input data table is completed. Table 3 presents the new, enriched dataset.

Table 3. An expert-extended version of Table 1 with a new, meaningful column d .

a	b	c	d	Decision
3	9	7	−3	0
1	4	4	0	1
5	9	4	16	2
⋮	⋮	⋮	⋮	⋮

The significance of parameter d is clear. Additionally, the classification rule becomes evident and can be manually replicated using a three-level relay-type function. This is confirmed by the results of the numerical experiments presented in Table 4 below.

Table 4. RMSE for selected ML models with KC.

	RMSE Train	NRMSE Train	RMSE Test	NRMSE Test
Linear Regression	0.4778	0.4639	0.5779	0.5837
Decision Tree	0.0000	0.0000	0.0000	0.0000
Random Forest	0.0408	0.0396	0.0333	0.0336

The error of the linear classifier decreases, but still, the model performance is not acceptable. The simple DT presented in Figure 4 classifies data with 100% accuracy and clearly outperforms the much more complex RF model. Now, LIME and SHAP explanations for the RF model that involves a KC are consistent, showing that parameter d has been identified as the most crucial determinant for the model's predictions, significantly outweighing the influence of parameters a , b , and c , whose contributions are not always consistent but are comparatively negligible. However, neither LIME nor SHAP explains why parameter d is the decisive parameter, i.e., the semantics behind it.

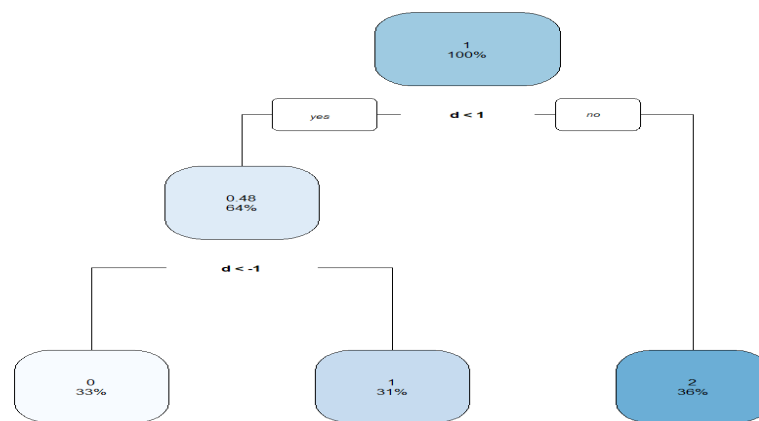


Figure 4. Decision tree model.

2.3. Explaining Grammatical Evolution

Grammatical Evolution (GE) is a form of Evolutionary Algorithm (EA) that mimics the process of biological evolution to seek solutions to problems [35]. Unlike a classical EA, GE integrates genetic algorithms with the rules of formal language theory. User-defined context-free grammars (CFGs) constrain the structure and syntax of the genome, which allows for the generation of solutions with a well-defined structure. The core idea of GE revolves around the process of mapping genotype to phenotype, which is achieved by generating sentences according to a specified grammar. In this process, the genome guides the selection of applicable production rules from the provided grammar. By imposing rules and restrictions through context-free grammars, GE ensures that the solutions adhere to specific syntactic patterns, enhancing their coherence and applicability to the problem domain and offering a framework for shaping and refining solutions, enabling the generation of outputs that align closely with desired specifications and objectives. Each solution is evaluated using a fitness value for a given objective function.

There are several implementations of GE algorithms. Notable examples include gramEvol for R; PyNeurGen, PonyGE, and PonyGE2 for Python; GEVA and ECJ for Java; and GELab for Matlab, among others (https://en.wikipedia.org/wiki/Grammatical_evolution#Implementations (accessed on 29 March 2024)).

While GE is used primarily to generate variable-length linear genome encodings of computer programs, it can also be employed for symbolic regression tasks and identification of functional dependencies. GE utilizes a CFG defined as $G = (N, T, P, S)$, where N is a set of non-terminal symbols, T is a set of terminal symbols, P consists of production rules, and S is the start symbol. Each individual in GE is represented by a chromosome (a string of integers that act as genetic material), guiding the selection of production rules from P to convert non-terminals to terminals, thus constructing valid mathematical expressions using a modulo operation based on available rule choices. The translation from genotype (chromosome) to phenotype (expression) involves applying the production rules based on gene values, iterating until a complete expression is formed, or the chromosome is exhausted. Fitness is evaluated by how closely these expressions match target outputs (e.g., with MSE). Evolutionary processes including crossover and mutation generate new chromosomes, with selection based on fitness, perpetuating the cycle to evolve increasingly accurate symbolic expressions.

The GE approach adds an extra layer of interpretability to the generated solutions. Since each solution must conform to the rules defined in the grammar, it is possible to trace back the decision-making process of the algorithm, providing insights into how specific solutions were constructed. While shallow XAI techniques such as LIME or SHAP can improve the understanding of black-box models by explaining their predictions in specific cases, GE helps in creating inherently interpretable models. GE-based explanations are directly derived from the model itself, without the need for approximation or additional

modeling, which may reduce the errors typically associated with the approximations made by post hoc explanations. The flexibility in how models are represented allows for more adaptable and potentially understandable models. The application of GE can be particularly advantageous in domains where the incorporation of expert knowledge into the model is crucial, such as in health care, as user-defined grammars allow for the incorporation of domain-specific constraints and guidelines. By optimizing multiple objectives (e.g., model accuracy, complexity, and interpretability), GE can support the development of predictive models that are both precise and understandable [36].

2.4. Application of Grammatical Evolution

We redefine the experiment and assume the domain expert has provided only numerical values of the variable d as a column of experimental results. However, leveraging domain-specific knowledge, the expert can offer insights into how d might depend on other variables. In this phase of the experiment, the `gramEvo1` package, a native R implementation of grammatical evolution, is utilized (it is also used in the subsequent experiments outlined in this paper). We maintain the use of 100 observations, as in the previous stages, but apply GE methods.

External knowledge can be represented as a context-free grammar and utilized as input for the GE process. The following is an example of such a grammar:

```
<expr> ::= <var> | <op>(<n>, <var>) | <op>(<var>, <n>)
        | <op>(<expr>, <expr>)
<op> ::= "-" | "*" | "^"
<var> ::= a | b | c
<n> ::= 1 | 2 | 3 | 4
```

Since GE is capable of performing symbolic regression tasks, it can identify functional dependencies within the data. The following expression for the d variable was derived utilizing the GE approach after fewer than 30,000 iterations:

Best Expression: $b^2 - c * 4 * a$

Using the derived formula for d , an additional component is incorporated into the model. The three-class classification problem is simplified into three distinct binary classification cases. Subsequently, a set of rules is established to determine the number of roots for a quadratic equation tailored specifically to each case:

```
ifelse(d > 0, "two", "other no. of roots")
ifelse(d == 0, "one", "other no. of roots")
ifelse(d < 0, "zero", "other no. of roots")
```

The result of combined rules is a simple and human-interpretable classification rule:

```
ifelse(d > 0, "two", ifelse ( d == 0 , "one", "zero"))
```

It should be noted that although $d > 0$, $d == 0$, or $d < 0$ is always included in the resulting formula of the GE process, occasionally, more complex rules, e.g., `ifelse((b > min(b)) & (d == 0), "one", "other no. of solutions")` are proposed as the final solution.

As might be expected, the classification model developed using the GE approach achieves 100% accuracy.

The classification explanation is given by the functional structural diagram shown in Figure 5. This diagram includes two functional components; the first calculates the value of d , and the second determines the actual class based on the value of d .

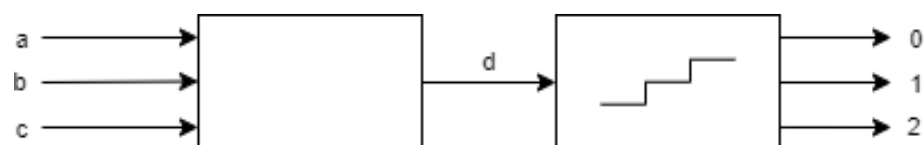


Figure 5. The functional model.

Note that in general, the diagram can take a more complex form of a directed acyclic graph. A more precise statement of the underlying theory is presented in a subsection below.

Incorporating external knowledge components (auxiliary domain knowledge and expert knowledge) can often enhance the levels of explainability and trustworthiness of AI models. A well-fitting KC can improve interpretability, address ethical issues, provide context, and boost model performance. Aligning the decision-making process with established principles or rules makes it easier to comprehend and explain the model's predictions. By incorporating domain knowledge, logical constraints, or expert guidelines, models can propose explanations that align with human reasoning. Additionally, domain expertise allows the model to capture underlying patterns and relationships that may not be apparent in the training data alone. This may result in better coverage and generalization of new, previously unseen data.

While incorporating external knowledge into the explanation process can be advantageous, it is crucial to verify the reliability and quality of the external source. Additionally, potential biases or limitations in the external information should be carefully considered [34].

3. Results

3.1. A Conceptual Framework for Model-Driven Explainable Artificial Intelligence

In this section, we briefly explain the concept of model-driven XAI (or model-based XAI), somewhat in opposition to shallow XAI approaches such as LIME or SHAP. The concept of causality plays the principal role in MD-XAI, in addition to causal reasoning, causal models, causal graphs, etc. [23]. The phenomenon of causality is a core mechanism to explain the behavior of systems. It can be used both forwards—by a kind of deductive reasoning or forward chaining—for predicting the results of certain input and backwards for a kind of abductive reasoning or backward chaining—for finding potential causes of some observed system outputs. Both forward reasoning and backward reasoning can be multistep procedures based on a graphical framework constituted by a multi-level graph. In [37], we presented a probabilistic causal model for biomedical data. It consists of a DAG with clear presentation of the causal dependencies among variables. Here, we consider a similar framework but a strictly deterministic one.

Here, we put forward the idea of the “3C principle”. In order to build an explainable model, we propose that the following 3 concepts be used in common:

- Causality —identification of causal dependencies among variables;
- Components —identification of system components; and
- Connections —identification of the connections of inputs, components, and outputs.

The abovementioned picture comes from the ideas of systems science and automatic control. A good example of such knowledge representation is the multiplier–adder circuit introduced in the Reiter’s famous paper on model-based diagnosis [25]. Some other examples used in ML/AI are Bayesian Networks (BNs) [23,37].

In order to provide clear intuitions for further discussion, let us briefly refer to the example network discussed in Section 2.1. It is the example of the famous multiplier–adder system presented in detail in the seminal paper on model-based diagnosis [25]. In order to focus on explanatory reasoning, we can follow the mode of Ligęza [30]. This example clearly explains the intuitions behind the following concepts:

- System inputs (A, B, C, D, and E);
- System outputs (F and G);
- Meaningful intermediate variables (X, Y, and Z);
- System components (three multipliers and two adders);
- Causality—the input signals and the defined work of components that cause the actual values of the intermediate and output variables.

For MD-XAI, we need an extended input; apart from a data table, we also need to incorporate some knowledge, e.g., the components, connections, intermediate values, etc.

Definition 1. Let $A = \{A_1, A_2, \dots, A_n\}$ be a set of attributes denoting the input variables, and let D be the output or decision variable. A table (T) of data specifications for a machine learning problem takes the following form presented in Table 5:

Table 5. A typical data table for ML.

A_1	A_2	$\dots$	A_n	D
$a_{1,1}$	$a_{1,2}$	$\dots$	$a_{1,n}$	d_1
$a_{2,1}$	$a_{2,2}$	$\dots$	$a_{2,n}$	d_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$a_{m,1}$	$a_{m,2}$	$\dots$	$a_{m,n}$	d_m

The basic task of ML is to induce a decision tool or procedure capable of predicting the value of the decision column on the basis of the data columns.

The proposed approach consists of the following steps:

- Selecting existing values or injecting new meaningful intermediate values;
- Determining the functions defining their values;
- Incorporating composition boxes;
- Defining connections among composition boxes;
- Evaluating the parameters (such as simplicity, error rate, etc.);
- Defining the exceptions uncovered by the model,
- Final simplification or modification/tuning of the structure and parameters.

The basic concept to be introduced is that of Meaningful Intermediate Variable (MIV), which is defined as follows:

Definition 2. A meaningful intermediate variable (MIV) is a (new) variable satisfying the following conditions:

- It has clear semantics; knowledge of its value is of some importance to the domain experts, and it can be understood by the data analyzers;
- Its values are defined with a relatively simple function (for example, from a set of predefined, admissible base functions);
- The input for the function is defined by the data table or other previously calculated MIVs.

A typical MIV defines a new, meaningful parameter that can be easily interpreted by the users and that moves the discussion to an upper, abstract level. The function behind an MIV can typically have 2–3 inputs. Therefore, the role of introducing MIVs is threefold, including the following:

- Reduction of the problem dimension by combining several inputs (from the initial dataset) into a single, newly created meta-level parameter;
- Catching the causal and functional dependencies existing in the dataset;
- Moving the explanatory analysis to a higher, abstract level in order to simplify it.

Typical examples of MIVs include introducing/calculating the area of specific surface, volume, density, ratio of coexisting values, etc. In the introductory example, it was the d variable, as introduced in the 4th column of Table 3.

A perfect example from the medical domain is the *Body Mass Index (BMI)*, which is defined as follows:

$$BMI = \frac{W}{H^2},$$

where W is the weight of a person (in kilograms) and H is his height (in meters). The *BMI* parameter is a MIV, while W and H are input data values. An experiment concerning the explanation of BMI considering LIME and SHAP versus grammatical evolution was reported in [38].

Note that the definition of MIVs constitutes an *extra knowledge component* and is not included in the typical dataset from ML repositories. Invention and injection of MIVs may correspond to a tradeoff between random selection from the base of available MIVs and expert advice based on strong domain knowledge.

Also note that MIVs can be organized into several levels, and the outputs of some of them may serve as inputs for others.

3.2. Explanation Examples: IRIS

In this section, we consider several white-box, model-driven XAI models for the widely explored IRIS dataset. The IRIS dataset consists of 150 instances, where each instance represents an iris flower sample. Each sample has four features/attributes, namely sepal length, sepal width, petal length, and petal width. Based on these features, each sample in the dataset is labeled with one of three classes, representing different species of iris flowers, namely *Setosa*, *Versicolor*, or *Virginica*.

The dataset is balanced, with 50 samples for each of the three species. Moreover, it demonstrates good separability among the different species of iris flowers. The simplicity and small size of the dataset make it a popular choice for algorithm testing in the field of ML.

Initially, we applied the GE approach to four features from the dataset (minimizing the number of misclassified labels, allowing 1000 iterations, and with default parameters), utilizing the following grammar:

```
expr = grule((expr) & (sub.expr),
            (expr) | (sub.expr),
            sub.expr),
sub.expr = grule(comparison(var, func.var)),
comparison = grule('>', '<', '==', '>=', '<='),
func.var = grule(num, var, func(var)),
func = grule(mean, max, min, sd),
var = grule(Sepal.Length, Sepal.Width, Petal.Length, Petal.Width),
num = grule(1, 1.5, 2, 2.5, 3, 4, 5)
```

Below, we present illustrative examples that demonstrate the separability of the dataset, including the resultant formulas and plots for the *Setosa* (Figure 6) and *Versicolor* (Figure 7) classes:

```
ifelse(Petal.Length <= Sepal.Width, "setosa", "other")
```

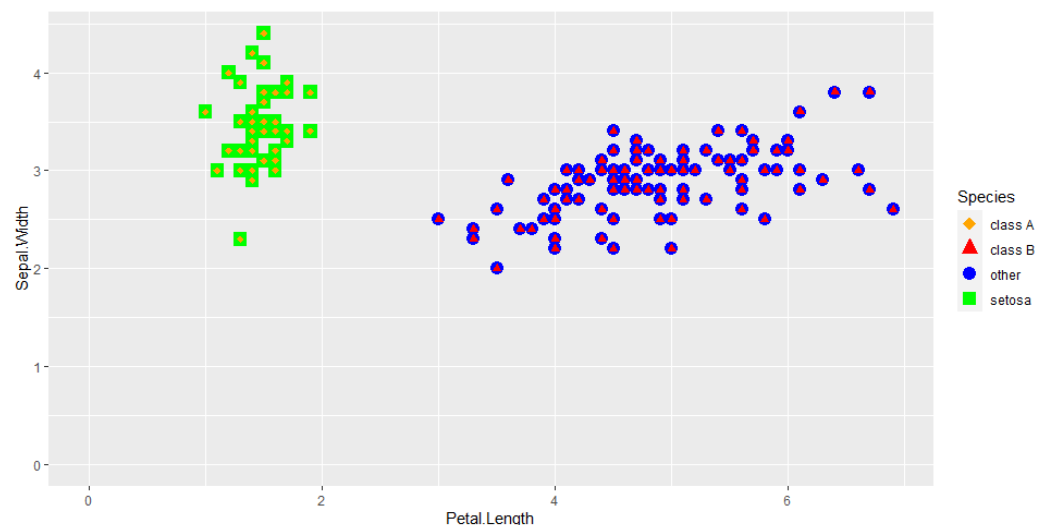


Figure 6. *Setosa* species classification using GE with sepal width and petal length parameters.

```
ifelse((Petal.Width <= 1.5) & (Petal.Length >= 2.5), "versicolor", "other")
```

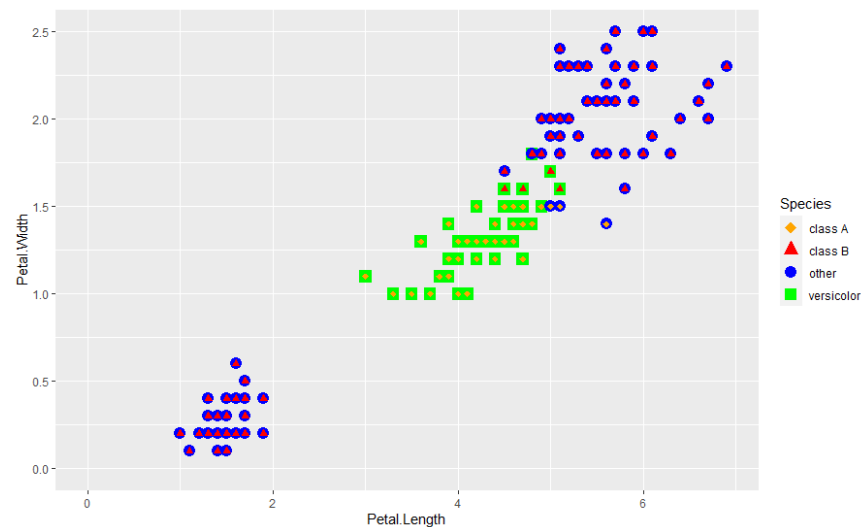


Figure 7. *Versicolor* species classification using GE with petal width and petal length parameters.

While the classification accuracy for the *Setosa* species was perfect, the accuracy for the other classes suggests potential for improvement (e.g., 94.66% for *Versicolor*).

To estimate confidence intervals (CI) for the accuracy of rule-based classification models, which traditionally lack direct uncertainty measures due to their deterministic nature, we applied a bootstrap resampling technique. Utilizing the *boot* package in R, we performed 1000 resamplings of the IRIS dataset. Each sample was evaluated using a predefined model that classifies instances as a given species or other based on fixed thresholds of the input features. This process of recalculating the model accuracy for each bootstrap sample generated a distribution of accuracies. From this distribution, percentile-based confidence intervals were derived, quantifying the variability and reliability of the model performance across different samples. The estimated CI (with a 95% confidence level) for *Versicolor* is (0.9067, 0.9800).

To improve the classification, we introduced MIVs.

In order to maintain simplicity, several models are presented in order but according to the same, simple scheme.

- Causal model—data flow;
- Input, output, and MIVs;
- Functional model;
- Accuracy;
- Simplicity.

The first model is very simple; only one MIV is introduced, and it is a value intended to be proportional to the petal area.

The inputs are *petal.length* and *petal.width*

$$PLmPW = \text{petal.length} * \text{petal.width} \quad (1)$$

We allowed for MIVs in the grammar definition. For each species, we enabled 1000 iterations of the evolutionary process. The complete functional model introduces a simple comparison of the MIV value defined by Equation (1) with the threshold values presented below.

```
ifelse(PLmPW <= 2, "setosa", "other")
ifelse((PLmPW <= 8) & (PLmPW >= 2), "versicolor", "other")
ifelse(PLmPW >= 8, "virginica", "other")
```

For intuition, in human-understandable terms, *PLmPW* can be interpreted as the “rough area” of a petal. The rules can be put together to create the functional model shown in Figure 8.

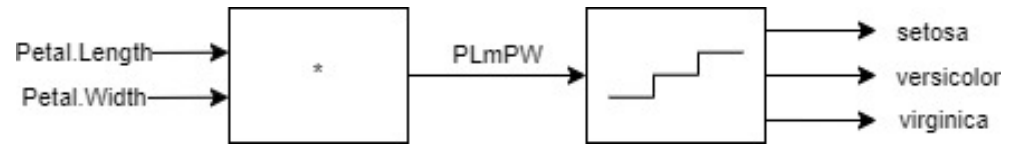


Figure 8. Functional model for iris classification with an MIV .

The results of the classification process performed with the provided functional model are depicted in Figure 9.

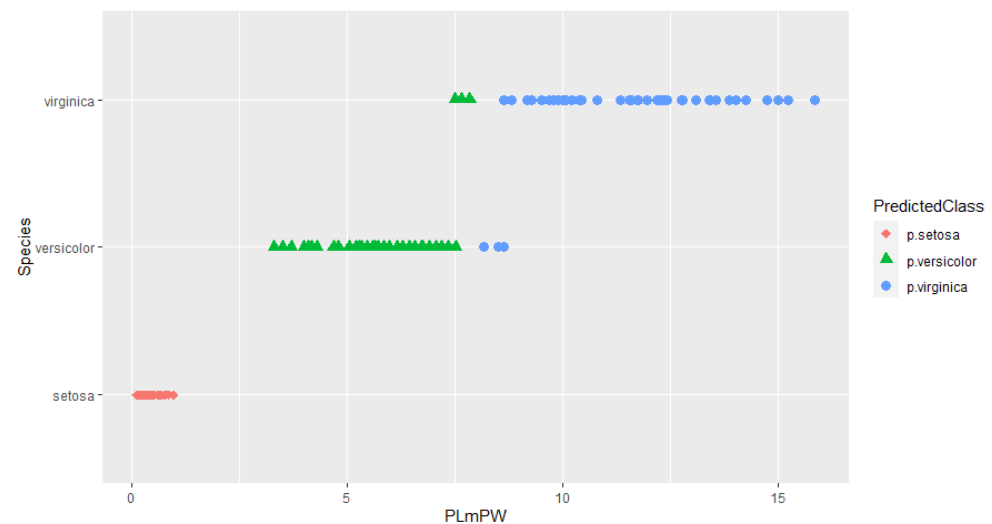


Figure 9. Three-class classification using a functional model with $MIV = petal.length * petal.width$.

The classification model, although not perfect, achieves a relatively high accuracy of approximately 95%. The model exhibits a high degree of simplicity, utilizing merely two input variables and a single MIV, with the number of outliers limited to only seven. The model is interpretable and easy to understand for humans. The 95% CI for the model accuracy is (0.9200, 0.9867).

Functional models can incorporate more MIVs.

Now, let us introduce another MIV that estimates sepal area. The inputs are sepal.length and sepal.width

$$SLmSW = sepal.length * sepal.width \quad (2)$$

MIVs in the model can be derived not only from input parameters but also based on other MIVs. An example of such an MIV is the petal–sepal area ratio based on MIVs (1) and (2).

$$PdS = PLmPW / SLmSW \quad (3)$$

Utilizing the GE approach, the following rules were generated based on the MIV value defined by Equation (3):

```
ifelse(PdS <= 0.2, "setosa", "other")
ifelse((PdS <= 0.43) & (PdS >= 0.22), "versicolor", "other")
ifelse(PdS >= 0.43, "virginica", "other")
```

The rules come together to form the functional model shown in Figure 10.

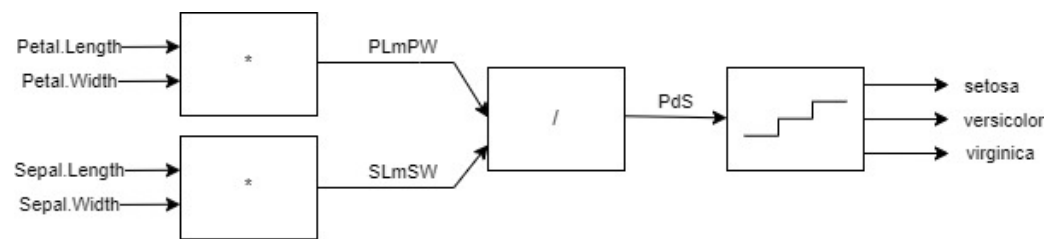


Figure 10. Functional model for iris classification with multiple MIVs .

The outcomes of the classification executed using the given functional model are shown in Figure 11.

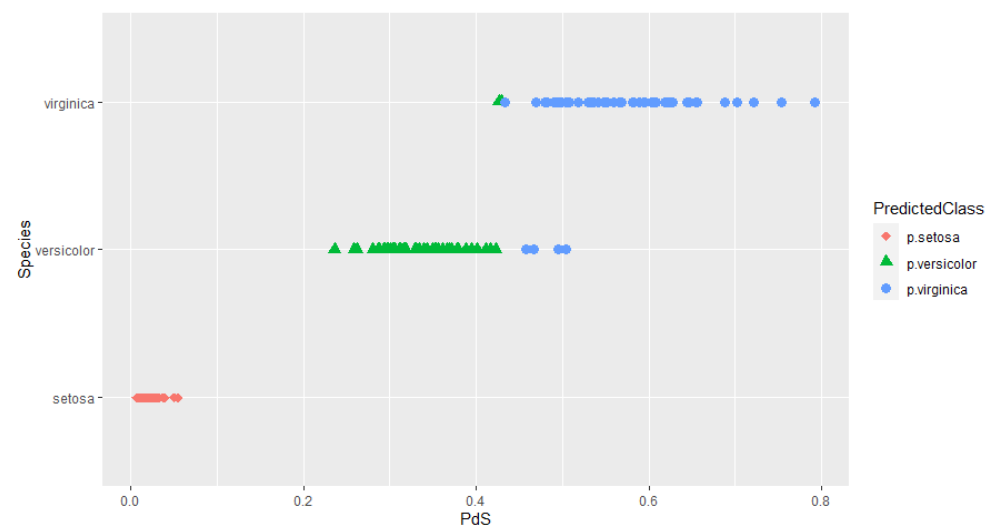


Figure 11. Three-class classification using a functional model with multiple MIVs.

The model shows slight improvement compared to the previous one. The number of outliers is reduced to six (accuracy improved to 96%). The 95% CI for the model accuracy is (0.9267, 0.9867). The model is more complex than its predecessor yet remains interpretable to humans.

Furthermore, in order to statistically compare the functional models shown in Figure 8 (Model 1) and Figure 10 (Model 2), we performed analysis of variance (ANOVA) on 1000 bootstrap samples for both models, specifically examining their respective accuracies. The complete results of ANOVA are presented in Table 6. A low p -value ($< 2 \times 10^{-16}$) indicates that there is a statistically significant difference in accuracy between the two tested models. The significant F value (96.75) suggests a strong model effect on accuracy.

Table 6. ANOVA results.

	Degrees of Freedom	Sum of Squares	Mean Square	F-Value	p -Value
Model 2 vs. Model 1	1	0.0278	0.027776	96.75	$< 2 \times 10^{-16}$
Residuals	1998	0.5736	0.000287		

As ANOVA indicated significant differences between models, we performed Tukey's Honestly Significant Difference (HSD) test. The adjusted p -value (effectively zero) indicates that the difference in mean accuracy between Model 1 and Model 2 is statistically significant. Model 2 has a higher mean accuracy than Model 1 by approximately 0.745%, with a 95% CI of (0.0060, 0.0089).

4. Discussion

While GE is primarily used in the field of genetic programming, it can also be relevant in the area of XAI due to its ability to generate human-readable and interpretable solutions that resemble the human way of thinking. Grammars can restrict the structure and components of possible explanations and indicate causality. GE can provide functional representations of the model that mimics the decision-making process. These representations can be analyzed by humans in order to gain insights into the model structure and way of working and, thus, to allow for better understanding of the behavior of the AI system.

The presented experiments point to the necessity of including more complete external declarative knowledge in building reliable and trustworthy AI models. The KC can help to provide context information and include domain-specific expertise. By incorporating this knowledge, models can better capture the problem and provide better predictions. Incorporating external knowledge can enhance the model's robustness against uncertainties, incomplete information, noisy data, and hidden nonlinearities. Moreover, the use of expert knowledge can aid in developing solutions that are robust and generalize well to new data and diverse scenarios.

One of the major disadvantages of the GE approach is the problem with convergence and reproducibility of the results. A GA does not guarantee finding optimal/any solutions in all cases. The search space of many real-world problems is vast and complex, making it difficult to guarantee finding the optimal solution. Selected experiments show that different runs of GE algorithms can return different functional models as explanations, especially for classification problems. Moreover, despite the demonstrated efficacy of grammatical evolution, the method can sometimes become trapped in local minima of the objective function, leading to performance that may fall short of expectations. To address these challenges, hybrid techniques combining local minimization algorithms with grammatical evolution can be employed. However, as grammatical evolution involves integer optimization, these techniques must be specifically tailored to suit this framework [39].

Context-free grammars can be employed to constrain the structure of an explanation model. However, a growing demand for penalization of complexity and redundancy within a model has arisen. Currently, there are no mechanisms to verify if the proposed genome includes superfluous information, making it challenging to eliminate unnecessary components. Additionally, estimating the number of building blocks necessary for a functional model a priori remains difficult.

At this moment, numerical threshold optimization is not performed effectively by the GE algorithm and is strongly dependent on user-specified numerical values. Future works should involve the implementation of solutions for optimal cutoff point searches.

Ensuring ethical and legal compliance necessitates the incorporation of external knowledge. Models must align with ethical guidelines, legal regulations, and privacy standards. By including an external knowledge component that addresses these elements, models can be constructed focusing on reliability and responsible decision making.

Our proposal introduces a novel application of GE in the area of XAI called model-driven explainable artificial intelligence. This approach transcends the limitations of traditional 'black-box' AI models by providing a transparent, interpretable framework that resembles human reasoning. By using GE for function identification, we not only enhance the technical capabilities of AI but also bring it closer to human thinking patterns, making AI decisions more understandable and trustworthy.

In the proposed approach, i.e., MD-XAI, there is a visible tradeoff between the simplicity and interpretability of the model and its accuracy. Our point is that it may be reasonable and desired to sacrifice the accuracy of the prediction in order to obtain a simpler, transparent, and robust model composed of easy-to-understand variables (MIVs), components, and functions. The number of building blocks of a functional model must be limited by human perceptual capabilities, as overly complex models, although accurate, may become incomprehensible to users. The accuracy of technical systems is never perfect, while understanding the real nature of the phenomena is always an ultimate goal.

Understanding functional dependencies in relational database systems (RDBSs) is critical for more than just technical reasons. It involves knowing the specific needs of different fields, making sure systems can handle unexpected changes and work with new data. Following ethical rules and laws is also of primary importance for building trust with users. Importantly, having experts check and assess the quality of these systems ensures they are reliable and useful and meet legal standards. This approach makes RDBSs not only more effective but also trusted sources of information in our data-driven world.

Our experiments present several challenges, particularly in the areas of dataset characterization, threshold optimization, and penalization of complexity and redundancy. An intrinsic limitation arises from the need to carefully define the characteristics of the target datasets, which is a prerequisite for tuning the operational parameters of the GE, including the optimization of thresholds that balance sensitivity and specificity. In addition, the penalty mechanism for complexity and redundancy is critical to refining the efficiency of the model but introduces additional layers of decision complexity. A major concern within this framework is the issue of GE convergence and reproducibility of results. The inherent variability in GE processes often prevents consistent results, raising questions about the stability and determinism of the approach. While this variability reflects the dynamic nature of evolutionary algorithms, it necessitates a deeper investigation into methods that could improve the predictability and reliability of results, thereby moving towards a more deterministic model. Addressing these challenges is critical to advancing the application of GE in model-driven XAI, ensuring that models not only capture the nuanced relationships within the data but also do so with high degrees of reliability and reproducibility.

5. Conclusions

The presented experiments demonstrate that relying solely on shallow explainability techniques can result in incorrect models and misleading or inconsistent explanations. The discrepancies in explanations highlight the inherent limitations of the abstract, model-agnostic approach. Whenever feasible, simple models and explanation techniques that foster a deep understanding of the model's structure and behavior should be preferred.

A promising technology for advancing and expanding the identification of interpretable functional structures is grammatical evolution. GE can be effectively used to identify functional dependencies within data. The presented experiments highlight the benefits of generating an accurate model with a small amount of training data compared to shallow methods, showcasing its potential in creating interpretable AI models and enhancing transparency. Models created using GE, even with limited amounts of training data, can surpass those developed by more traditional, shallow techniques. The adaptability of GE to different data types and its capacity to reveal complex patterns underline its robustness and versatility.

This study underscores the importance of the 3C principle (Causality, Components, and structure) in enhancing the interpretability and effectiveness of model-driven XAI. This principle serves as a guideline for the construction of transparent and understandable AI models. By emphasizing causality, identifying crucial system components, and elucidating their interconnections, the 3C principle facilitates a deeper understanding of the underlying mechanisms of AI decisions. Adopting this principle not only advances the development of explainable models but also helps to build trust and reliability in AI systems.

The significance of meaningful intermediate variables is also paramount in the construction of explainable AI models. MIVs, by providing an additional layer of abstraction, help in reducing the complexity of data models while retaining essential information that contributes to the interpretability of the AI decision-making process. These variables enable the encapsulation of complex relationships within the data into more understandable and tangible concepts, thereby facilitating a more intuitive analysis and explanation of AI models. Collaboration between computational techniques and external knowledge (human expertise) is critical to the advancement of ethical and responsible AI, ensuring that models are aligned with real-world applicability and constraints. By bridging the gap

between raw data and high-level interpretations, MIVs play a crucial role in demystifying AI operations and enhancing the accessibility of AI insights for domain experts and other users of AI systems.

This study contributes to the field of XAI by offering a novel approach that bridges the gap between the predictive power of machine learning and the human need for understanding and trust in AI systems. The insights gained herein pave the way for future research efforts aimed at creating AI models that are not only powerful but also transparent and in line with ethical standards.

Future work should focus on enhancing the efficiency and scalability of the MD-XAI approach. One critical aspect under consideration could be the reduction of computational complexity, which remains a significant challenge with respect to the broader adoption and application of these methodologies. Efforts could be directed towards optimizing the grammatical evolution process and the selection of MIVs to ensure that models can be developed and interpreted with reduced computational resources, making the technology more accessible for large-scale and real-time applications.

Author Contributions: Conceptualization, D.S. and A.L.; methodology, D.S. and A.L.; software, D.S.; validation, D.S. and A.L.; formal analysis, D.S. and A.L.; investigation, D.S. and A.L.; resources, D.S.; data curation, D.S.; writing—original draft preparation, D.S. and A.L.; writing—review and editing, D.S. and A.L.; visualization, D.S.; supervision, A.L.; project administration, A.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ML	Machine Learning
AI	Artificial Intelligence
XAI	eXplainable Artificial Intelligence
LIME	Local Interpretable Model-Agnostic Explanations
SHAP	SHapley Additive exPlanations
DAG	Directed Acyclic Graph
LR	Linear Regression
DT	Decision Tree
RF	Random Forest
MD-XAI	Model-Driven XAI
KC	Knowledge Component
GE	Grammatical Evolution
EA	Evolutionary Algorithm
CFG	Context-Free Grammar
CI	Confidence Interval
MIV	Meaningful Intermediate Variable
ANOVA	Analysis of Variance

References

1. Marwala, T. *Handbook of Machine Learning—Volume 1: Foundation of Artificial Intelligence*; WorldScientific: Singapore, 2018. [\[CrossRef\]](#)
2. Caiafa, C.F.; Sun, Z.; Tanaka, T.; Marti-Puig, P.; Solé-Casals, J. Special Issue “Machine Learning Methods for Biomedical Data Analysis”. *Sensors* **2023**, *23*, 9377. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Dwivedi, A.K.; Dwivedi, M. A Study on the Role of Machine Learning in Natural Language Processing. *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.* **2022**, *8*, 192–198. [\[CrossRef\]](#)

4. Sarker, I.H. Machine Learning: Algorithms, Real-World applications and research Directions. *SN Comput. Sci.* **2021**, *2*, 160. [[CrossRef](#)] [[PubMed](#)]
5. Nassehi, A.; Zhong, R.Y.; Li, X.; Epureanu, B.I., Review of machine learning technologies and artificial intelligence in modern manufacturing systems. In *Design and Operation of Production Networks for Mass Personalization in the Era of Cloud Technology*; Elsevier: Amsterdam, The Netherlands, 2022; pp. 317–348. [[CrossRef](#)]
6. Cios, K.J.; Pedrycz, W.; Swiniarski, R.W. *Data Mining Methods for Knowledge Discovery*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012; Volume 458.
7. Cios, K.J.; Pedrycz, W.; Swinarski, R.W.; Kurgan, L.A. *Data Mining. A Knowledge Discovery Approach*; Springer Science: New York, NY, USA, 2007.
8. Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.* **2019**, *1*, 206–215. [[CrossRef](#)] [[PubMed](#)]
9. Magnani, L.; Bertolotti, T. *Springer Handbook of Model-Based Science*; Springer: Berlin/Heidelberg, Germany, 2017.
10. Arrieta, A.B.; Díaz-Rodríguez, N.; Del Ser, J.; Bannetot, A.; Tabik, S.; Barbado, A.; García, S.; Gil-López, S.; Molina, D.; Benjamins, R.; et al. Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI. *Inf. Fusion* **2019**, *58*, 82–115. [[CrossRef](#)]
11. Guidotti, R.; Monreale, A.; Ruggieri, S.; Turini, F.; Giannotti, F.; Pedreschi, D. A Survey of Methods for Explaining Black Box Models. *ACM Comput. Surv.* **2019**, *51*, 1–42. [[CrossRef](#)]
12. Schwalbe, G.; Finzel, B. A comprehensive taxonomy for explainable artificial intelligence: A systematic survey of surveys on methods and concepts. *Data Min. Knowl. Discov.* **2023**. [[CrossRef](#)]
13. Linardatos, P.; Papastefanopoulos, V.; Kotsiantis, S. Explainable AI: A Review of Machine Learning Interpretability Methods. *Entropy* **2021**, *23*, 18. [[CrossRef](#)] [[PubMed](#)]
14. Speith, T. A Review of Taxonomies of Explainable Artificial Intelligence (XAI) Methods. In Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, FAccT '22, Seoul, Republic of Korea, 21–24 June 2022; ACM: New York, NY, USA, 2022. [[CrossRef](#)]
15. Burkart, N.; Huber, M.F. A Survey on the Explainability of Supervised Machine Learning. *J. Artif. Intell. Res.* **2021**, *70*, 245–317. [[CrossRef](#)]
16. Adadi, A.; Berrada, M. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access* **2018**, *6*, 52138–52160. [[CrossRef](#)]
17. Ribeiro, M.T.; Singh, S.; Guestrin, C. “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, San Francisco, CA, USA, 13–17 August 2016; pp. 1135–1144.
18. Lundberg, S.M.; Lee, S.I. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; pp. 4765–4774.
19. Sepioło, D.; Ligeza, A. Towards Explainability of Tree-Based Ensemble Models. A Critical Overview. In *New Advances in Dependability of Networks and Systems*; Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J., Eds.; Springer: Cham, Switzerland, 2022; pp. 287–296.
20. Sepioło, D.; Ligeza, A. A Comparison of Shallow Explainable Artificial Intelligence Methods against Grammatical Evolution Approach. In *Progress in Polish Artificial Intelligence Research 4*; Lodz University of Technology Press: Lodz, Poland, 2023; pp. 89–94.
21. Mei, Y.; Chen, Q.; Lensen, A.; Xue, B.; Zhang, M. Explainable Artificial Intelligence by Genetic Programming: A Survey. *IEEE Trans. Evol. Comput.* **2023**, *27*, 621–641. [[CrossRef](#)]
22. Wang, Y.C.; Chen, T. Adapted techniques of explainable artificial intelligence for explaining genetic algorithms on the example of job scheduling. *Expert Syst. Appl.* **2024**, *237*, 121369. [[CrossRef](#)]
23. Pearl, J. *Causality. Models, Reasoning and Inference*, 2nd ed.; Cambridge University Press: New York, NY, USA, 2009.
24. Ligeza, A.; Jemioło, P.; Adrian, W.T.; Ślaziński, M.; Adrian, M.; Jobczyk, K.; Kluza, K.; Stachura-Terlecka, B.; Wiśniewski, P. Explainable Artificial Intelligence. Model Discovery with Constraint Programming. In *Studies in Computational Intelligence, Proceedings of the Intelligent Systems in Industrial Applications, 25th International Symposium, ISMIS 2020, Graz, Austria, 23–25 September 2020*; Selected Papers from the Industrial Part; Stettinger, M., Leitner, G., Felfernig, A., Raś, Z.W., Eds.; Springer: Berlin/Heidelberg, Germany, 2020; Volume 949, pp. 171–191. [[CrossRef](#)]
25. Reiter, R. A Theory of Diagnosis from First Principles. *Artif. Intell.* **1987**, *32*, 57–95. [[CrossRef](#)]
26. Hamscher, W.; Console, L.; de Kleer, J. (Eds.) *Readings in Model-Based Diagnosis*; Morgan Kaufmann: San Mateo, CA, USA, 1992.
27. Ligeza, A.; Kościelny, J.M. A new approach to multiple fault diagnosis. Combination of diagnostic matrices, graphs, algebraic and rule-based models. The case of two-layer models. *Int. J. Appl. Math. Comput. Sci.* **2008**, *18*, 465–476. [[CrossRef](#)]
28. Ligeza, A. A Constraint Satisfaction Framework for Diagnostic Problems. In *Diagnosis of Processes and Systems*; Control and Computer Science; Information Technology, Control Theory, Fault and System Diagnosis; Pomeranian Science and Technology Publisher PWNT: Gdańsk, Poland, 2009; pp. 255–262.

29. Ligeza, A. Towards Constructive Abduction: Solving Abductive Problems with Constraint Programming. In Proceedings of the International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K, Lisbon, Portugal, 12–14 November 2015; Volume 2, pp. 352–357.
30. Ligeza, A. An Experiment in Causal Structure Discovery. A Constraint Programming Approach. In *Lecture Notes in Computer Science, Proceedings of the Foundations of Intelligent Systems—23rd International Symposium, ISMIS 2017, Warsaw, Poland, 26–29 June 2017*; Kryszkiewicz, M., Appice, A., Slezak, D., Rybinski, H., Skowron, A., Raś, Z.W., Eds.; Springer: Berlin/Heidelberg, Germany, 2017; Volume 10352, pp. 261–268. [\[CrossRef\]](#)
31. Miller, T. Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.* **2019**, *267*, 1–38. [\[CrossRef\]](#)
32. Ali, S.; Abuhmed, T.; El-Sappagh, S.; Muhammad, K.; Alonso-Moral, J.M.; Confalonieri, R.; Guidotti, R.; Del Ser, J.; Díaz-Rodríguez, N.; Herrera, F. Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence. *Inf. Fusion* **2023**, *99*, 101805. [\[CrossRef\]](#)
33. Yu, K.; Li, J.; Liu, L. A review on algorithms for constraint-based causal discovery. *arXiv* **2016**, arXiv:1611.03977.
34. Ligeza, A.; Sepioło, D. In Search for Model-Driven eXplainable Artificial Intelligence. In *Artificial Intelligence for Knowledge Management, Energy and Sustainability*; Mercier-Laurent, E., Kayakutlu, G., Owoc, M.L., Wahid, A., Mason, K., Eds.; Springer: Cham, Switzerland, 2024; pp. 11–26.
35. Ryan, C.; O'Neill, M.; Collins, J.J. (Eds.) *Handbook of Grammatical Evolution*; Springer: Berlin/Heidelberg, Germany, 2018. [\[CrossRef\]](#)
36. Hu, T. Can Genetic Programming Perform Explainable Machine Learning for Bioinformatics? In *Genetic and Evolutionary Computation*; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; pp. 63–77. [\[CrossRef\]](#)
37. Fuster-Parra, P.; Tauler, P.; Bennasar-Veny, M.; Ligeza, A.; López-González, A.A.; Aguiló, A. Bayesian network modeling: A case study of an epidemiologic system analysis of cardiovascular risk. *Comput. Methods Programs Biomed.* **2016**, *126*, 128–142. [\[CrossRef\]](#)
38. Sepioło, D.; Ligeza, A. Towards Model-Driven Explainable Artificial Intelligence. An Experiment with Shallow Methods Versus Grammatical Evolution. In *Proceedings of the Artificial Intelligence. ECAI 2023 International Workshops*; Springer: Cham, Switzerland, 2024; pp. 360–365.
39. Tsoulos, I.G.; Tzallas, A.; Karvounis, E. Using Optimization Techniques in Grammatical Evolution. *Future Internet* **2024**, *16*, 172. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.