Artificial Intelligence for Science

Sašo Džeroski[0000-0003-2363-712X]

Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
Saso.Dzeroski@ijs.si

Abstract. Artificial intelligence is transforming science across many disciplines, yet realising its full potential requires addressing challenges specific to scientific work: models must be explainable and interpretable, capable of learning from the small labelled datasets typical in science, able to integrate existing domain knowledge, and facilitate representing and sharing scientific findings in an open and reproducible way. This chapter presents a range of AI methods and approaches developed to address these challenges. We cover explainable machine learning — with decision trees, rule sets, and their ensembles for multi-target prediction as key examples — along with semi-supervised learning for limited labelled data. We then discuss foundation models and how they can be adapted to scientific tasks through fine-tuning and transfer learning. Automated scientific modelling, in which interpretable systems of differential equations are learned from time series data, is presented next, followed by semantic technologies and ontologies for open science. Representative applications in life sciences, environmental sciences, and materials science illustrate the methods in practice. The chapter concludes with a discussion of AI infrastructure for science, including AI factories, illustrated with the example of the Slovenian AI Factory (SLAIF).

Keywords: AI for science, explainable machine learning, foundation models, automated scientific modelling, semantic technologies, open science, AI factories

1 Introduction

Artificial intelligence is transforming science across virtually every discipline, and its future impact promises to be even greater. This is not merely a reflection of the current excitement surrounding generative AI and large language models. Science has always been shaped by the tools available to it: AI now offers powerful tools to find patterns in complex data, build predictive models, automate experimental workflows, and represent and share scientific knowledge at scale.

At the same time, science exerts specific demands on AI that are not always met by off-the-shelf AI methods. To be useful in science, the outputs of AI methods must be understandable to domain scientists. Further, the AI methods need to be able to learn from limited amounts of (labelled) data, on one hand, and existing domain knowledge, on the other hand. Finally, their outputs must be reproducible. A large part of our research has been devoted to developing AI approaches that meet these demands: The present chapter gives an overview of the relevant areas and some highlights of our research, both methodological and applied.

The history of AI for science is considerably longer than the current public discourse might suggest. It begins in the late 1960s with DENDRAL, an expert system developed at Stanford for the interpretation of mass spectrometry data in organic chemistry —an early example of AI applied to a genuine scientific problem [24, 84]. A decade later, the BACON system demonstrated that computers could rediscover empirical laws such as Kepler's third law directly from numerical data [71, 76], inaugurating the field of computational scientific discovery. By the early 2000s, this community had accumulated substantial experience and distilled it into a set of principles that remain as relevant today as when they were first articulated [34, 74]. Scientific AI systems should produce outputs that are easy to communicate to domain scientists; they should exploit existing background knowledge rather than treating every problem as a blank slate; they should be able to learn from small datasets, since scientific experiments may be costly in time and labour; they should produce models that explain data rather than merely predict it; and they should support interaction with the domain scientists who use them.

Realising the potential of AI for science requires addressing several interconnected challenges. First, many scientific datasets are small by machine learning standards — a few hundred or a few thousand labelled examples, rather than the millions that underpin commercial AI applications. Second, domain knowledge is abundant in science, in the form of physical laws, biological constraints, or chemical principles, and failing to incorporate it wastes valuable resources or makes feasible tasks intractable. Third, science is an explanatory enterprise: a model that predicts well but cannot be interpreted is of limited value to a scientist seeking to understand a phenomenon. Fourth, the products of scientific work — data, models, protocols — must be represented in a form that is findable, accessible, interoperable, and reusable, both by humans and by machines.

Scientific research involves, on the one hand, data — obtained through observation, experimentation and simulation. On the other hand, it involves knowledge structures such as models, laws and theories [34]. AI for science must therefore support not only prediction from data, but also the construction, evaluation, representation and reuse of scientific knowledge.

This chapter presents a suite of AI methods developed with these challenges in mind, organised around four broad themes. Explainable machine learning encompasses methods that learn accurate yet interpretable (or explainable) models, with trees, rules, and their ensembles as central examples, inc. multi-target prediction and semi-supervised learning from limited labelled data. Foundation models provide a complementary response to limited labelled data, adapting broadly pretrained models to scientific tasks through zero-shot use, in-context learning, or fine-tuning. Automated scientific modelling focuses on the discovery of interpretable mathematical models, incl. systems of differential equations, directly from time-series data, augmented by domain knowledge. Semantic technologies for open science address the representation and sharing of scientific knowledge through ontologies and controlled vocabularies, supporting both reproducibility and automation of scientific workflows. We also present representative applications of these methods across scientific domains, i.e. life sciences, environmental sciences, and materials science, concluding with a discussion of the role of AI infrastructure in making AI capabilities accessible to the broader scientific community.

2 Explainable Machine Learning for Science

2.1 Explanation is Essential for Science

Modern machine learning models are often black boxes. A random forest [17] may contain hundreds of trees whose combined vote is not straightforward to unpack even when each individual tree is simple; a neural network may have millions of parameters whose nonlinear interactions resist inspection entirely. For science, this opacity is not a cosmetic inconvenience. Understanding *why* a model makes a given prediction is what allows a finding to be trusted, checked, and built upon. Explanation is at the very foundation of the scientific enterprise, not an optional add-on to it. A model that predicts accurately but cannot be interpreted is of limited value to a scientist. It is useful to distinguish two notions that are often conflated. An *interpretable* model is one whose internal structure can be understood by inspection alone, without recourse to any further method. Small decision trees, sparse linear models, classification rules, and naive Bayes classifiers are typical examples. Their structure can be inspected directly, provided that the learned model remains sufficiently small. An *explainable* model, by contrast, may remain a black box, but its behaviour can be probed after the fact through feature-importance estimates or local explanations of its predictions (with methods like LIME or SHAP, see Section 2.4). With an interpretable model, explanation comes for free, as a direct result of learning; with a black-box model, it comes separately, either as a by-product of learning (e.g., feature rankings can be produced as part of building a tree ensemble) or through a separate post-hoc procedure, at extra computational cost and a risk that the explanation is itself only an approximation.

Interpretability itself, moreover, is not free of complexity: a tree with hundreds of nodes or a linear model with thousands of coefficients is transparent in principle, but no easier to take in than a black box in practice, so an interpretable model is useful only when it stays genuinely small. Rudin [117] has argued that, in high-stakes applications, an interpretable model should be preferred outright over a black box paired with post-hoc explanation, a position this chapter is very much in sympathy with.

As deep neural networks have become increasingly prominent across the sciences without being able to explain their own predictions [150], explainable AI (XAI) [50] has emerged as a separate field in its own right. XAI work concentrates on the same two strands just distinguished: models interpretable by construction, and post-hoc methods for explaining and visualising black-box predictions after the fact [9, 49, 118]. Sections 2.2–2.3 discuss the former, while Section 2.4 discusses the latter.

One caution is worth stating up front. A post-hoc explanation describes the fitted model's behaviour, not necessarily the phenomenon the data represent — a variable can appear important through a measurement artefact or a proxy relationship rather than a genuine causal role, and correlated predictors can arbitrarily divide or exchange their apparent importance between them [92]. Predictive importance is not causal importance. For scientific use, XAI is best treated as part of a broader validation process: explanations should be checked for stability, compared across models, and weighed against domain knowledge. Ideally, they should lead to testable hypotheses.

2.2 From Single-target to Multi-target Prediction Tasks

Predictive modelling is the task most commonly addressed in machine learning. Its goal is to construct a model that predicts a target property of an object from its description (i.e., its other properties). The model is learned from examples consisting of descriptions and labels, i.e., values of the target property, and then applied to new, unseen cases, for which no labels are given. The model itself can take the form of an equation, a set of rules, a decision tree or another representation.

In single-target prediction, one target variable is predicted. If the target is discrete, the task is classification; if it is numeric, the task is regression. For example, predicting the (degree of) habitat suitability of a species from environmental variables is a regression task, while predicting a patient's diagnosis from laboratory measurements is a classification task. Classification and regression trees (CART) [16], also known as decision trees, are among the most widely used methods for both tasks. They combine good predictive performance with interpretability, as discussed in Section 2.1.

Predictive tasks can also be distinguished along two other dimensions. The first concerns the available supervision (i.e., labels). In fully supervised predictive modelling, all examples are labelled. In semi-supervised learning [142], only a part of the training set consists of labelled data, while the rest of the data is unlabelled. At the other end of the spectrum, unsupervised learning has no target variables (and thus the examples have no labels): It addresses clustering rather than prediction. Section 2.3 shows how predictive clustering connects these settings.

The second dimension concerns how the data become available. In batch learning, all examples are available before learning begins. In data-stream mining [45], examples arrive continuously and the model is updated incrementally. Such streams arise, for example, from automated laboratory instruments.

The most important dimension of organizing predictive modelling problems remains the type of targets considered. While single-target classification and regression have been predominant for decades, many scientific problems involve several related target variables. Multi-target prediction [151] addresses these variables jointly, using a single model. It can improve performance by exploiting dependencies among targets, rather than learning a separate model for each target. It can also provide a shared explanation of their variation. Much of our research over the past three decades has focused on developing methods for learning interpretable models for such tasks.

Figure 1 illustrates the difference between single- and multi-target prediction using data on patients with neurodegenerative diseases. The predictors are biomarkers derived from brain images, including glucose metabolism measured by FDG-PET, hippocampal volume, and amyloid deposition measured by AV45. Figure 1a) presents a single-target classification tree that predicts the patient's diagnostic category. It addresses one target and therefore answers one question.

The same patients are also assessed using clinical instruments that measure cognitive and functional impairment. These include the Clinical Dementia Rating Sum of Boxes (CDRSB) and the Alzheimer's Disease Assessment Scale (ADAS13). In the example considered here, 23 such clinical scores are recorded. They are related to one another and to the diagnosis: patients with more severe disease tend to perform worse on several

measures. Figure 1b) presents a multi-target regression tree (from a follow-up to Breskvar et al. [19]) that jointly predicts these 23 scores and the numerically encoded diagnosis from the imaging-derived biomarkers. Its splits reflect the joint variation of these targets and provide a common explanation of differences among the patients.

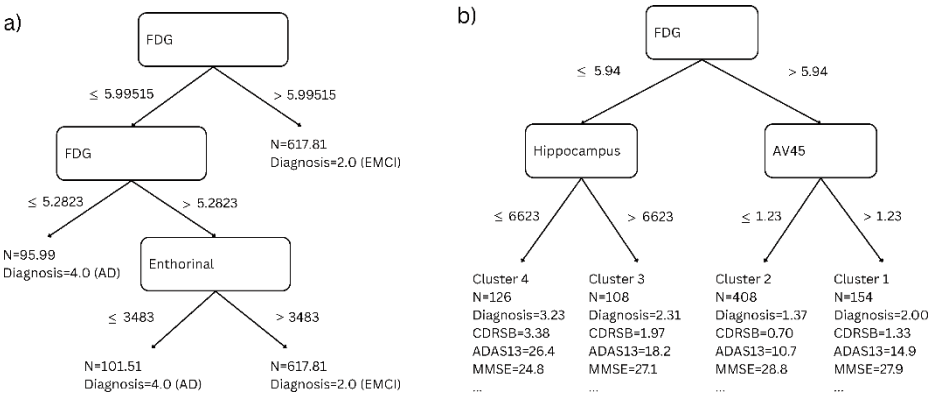

Fig. 1. Two decision trees relating features derived from brain images to clinical properties of patients with neurodegenerative diseases: **a)** a single-target classification tree predicting just the diagnosis and **b)** a multi-target regression tree jointly predicting diagnosis (numerically encoded) and multiple clinical scores (incl. CDRSB, ADAS13).

Multi-target prediction includes several task types. When all targets are numeric, the task is multi-target regression (MTR). When they are discrete, it is multi-target classification (MTC). Multi-label classification is the most important special case of MTC in which every target/ label is binary. Examples of MLC tasks include gene function prediction, where several functions can be assigned to a gene, and labeling images.

Hierarchical multi-label classification (HMLC) [146] is a special case of MLC, where the labels are additionally organised into a hierarchy or taxonomy. Predicting the structure of a community/ biota from environmental data, discussed in Section 6.2, is an example of this type of task: Here the models predict the organisms present at a river site at several taxonomic ranks, such as species, genus, family, and order.

The multi-target prediction tasks above differ in the type and organisation of their targets, but share the same underlying motivation. Related targets can be predicted more effectively and explained more coherently when they are modelled together. Predictive clustering trees provide a unified framework for addressing these tasks.

2.3 Predictive Clustering Trees: A Unifying Paradigm

Predictive clustering trees (PCTs), introduced by Blockeel et al. [10], generalise decision trees. A decision tree for classification or regression is typically constructed by top-down induction. At each internal node, the algorithm selects a test and partitions the examples according to its outcome. This is applied recursively to the resulting subsets until a stopping criterion is met, e.g., the target variance is low. Each leaf gives a prediction for the examples reaching it, e.g., the majority class or mean target value.

While splits are selected to maximise the reduction in target variance in decision trees, PCTs replace target variance with a more general measure of cluster variance. This measure can be defined over one or more target variables, the descriptive variables, or both. The same tree-induction algorithm can therefore address several types of tasks. These include single- and multi-target classification and regression, as well as supervised, semi-supervised and unsupervised learning.

Each leaf of a PCT represents a cluster of similar examples and provides a prediction for the examples in that cluster. The conditions along the path from the root to the leaf serve as a description and explanation of the cluster. This dual role makes PCTs interpretable. The tree in Figure 1b), for example, jointly predicts 24 targets. At the same time, its leaves identify groups of patients with similar clinical profiles, with the groups explained in terms of imaging-derived biomarkers.

The same framework supports both multi-target prediction and semi-supervised learning. When the target value is missing, the variance of the descriptive variables can still be calculated. A PCT can therefore be induced from both labelled and unlabelled examples. The split-selection criterion combines variance in the target space with variance in the descriptive space, with a parameter controlling their relative weight.

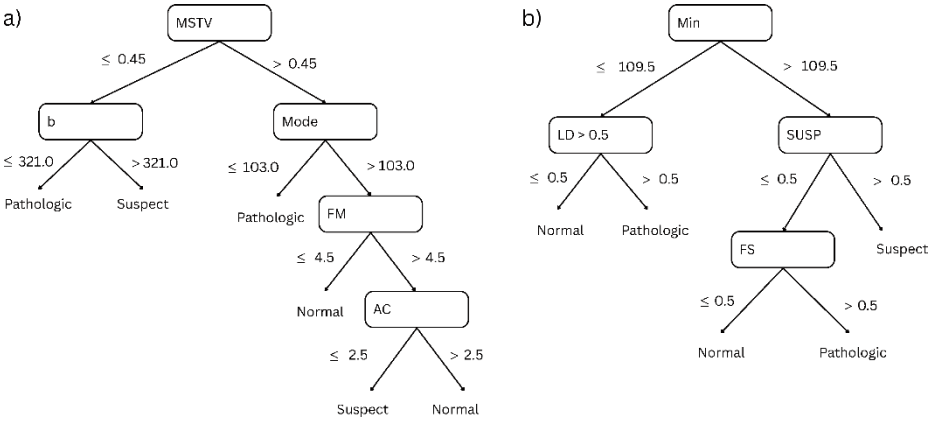

Fig. 2. Two classification trees learned from the same 50 labelled examples: **a)** A fully supervised tree learned from the labelled examples only. **b)** A semi-supervised tree learned using an additional 2,076 unlabelled examples. The first has 11 nodes and achieves 81% accuracy on unseen examples; the second has 9 nodes and achieves 92%.

Using both labelled and unlabelled data can result in models that are both accurate and simple enough to interpret. Figure 2 [78] illustrates this effect. The semi-supervised tree is smaller than the fully supervised tree and, at the same time, considerably more accurate. Note that the structure of the two trees differs significantly. The unlabelled examples provide information about the distribution of the values of the descriptive variables. This can prevent the tree from selecting splits suggested by a small labelled sample but unsupported by the broader structure of the data.

Over more than two decades, we have systematically extended PCTs along the three dimensions introduced above: the type of output, the degree of supervision, and batch versus stream learning. Concerning output, we started with PCTs for multi-target regression [131], then proceeded with multi-label and hierarchical multi-label classification [146]. Concerning supervision, we have covered the fully supervised and unsupervised settings, as well as semi-supervised learning for single-target classification and regression, multi-target regression, and (hierarchical) multi-label classification [79]. Most combinations of these two dimensions have been studied in the batch setting, with some extended to data streams. Our work has thus explored much of the Cartesian product defined by the three dimensions.

The batch methods are implemented in the open-source CLUSplus software [104]. CLUSplus provides a common implementation of PCTs for predicting different types of structured outputs. The streaming methods build on iSOUP-Tree, an incremental approach to building PCTs for multi-target regression implemented within the MOA framework [99], subsequently extended to MLC [98] and semi-supervised multi-target regression [96].

PCTs have also been extended from individual trees to ensembles [63]. Bagging and random forests construct multiple trees from different samples of the examples and variables, and combine their predictions. Such ensembles are usually more accurate and robust than individual trees, but are more difficult to interpret. Different ensemble mechanisms have also been used to estimate variable importance in multi-target prediction [106], including the symbolic, GENIE3 and permutation-based feature ranking approaches. In the latter, a variable is considered important if randomly permuting its values reduces predictive performance. Our approach provides rankings of the variables for predicting individual targets or all targets.

Predictive clustering can also be used to construct rules. Predictive clustering rules (PCRs) have the form *IF conditions THEN prediction*, where the prediction assigns values to the multiple targets. Unlike paths extracted from a tree, they can be learned directly and need not form a hierarchy. They can be organised as an unordered rule set or as an ordered decision list [156]. Each rule describes a cluster of examples and gives a prediction for that cluster. PCRs therefore retain the dual descriptive and predictive role of PCTs, while representing the model as a set of local descriptions.

PCTs have also been extended to relational data, where examples are described by multiple related objects rather than a single row in a table. TILDE, the original implementation of PCTs, already extended top-down tree induction to first-order logic, placing logical queries in the internal tree nodes [11]. Relational trees were subsequently extended with aggregate features, which summarise sets of related objects using functions such as count, minimum, maximum, sum and mean, and with ensemble functionality [141].

More recently, these ideas for learning relational trees and ensembles were implemented in Python within the software package RE3PY [109]. RE3PY constructs aggregate features from paths through relations and uses them in individual trees or ensembles. Ensemble mechanisms can also estimate the importance of relational features, identifying the relations, paths and aggregates that contribute most to the predictions.

2.4 Explaining Black-box Models and Their Predictions

Black-box models can be explained at two levels. Global explanations describe the behaviour of a model as a whole, whereas local explanations describe why the model made a particular prediction. The distinction is important: A variable that strongly influences the model overall need not be important for every individual prediction. Both types of explanation should also be distinguished from estimating the relevance of variables for a specified prediction task without reference to a particular fitted model.

Simple model-independent approaches rank variables according to their univariate association with the target. Such rankings may miss variables that are relevant through interactions with others. ReliefF estimates relevance by comparing neighbouring examples with similar and different target values [66]. It can detect variables that are useful in combination with others without inducing a predictive model. We have extended Relief-based feature ranking to multi-target prediction and high-dimensional data, specifically by using manifold embeddings in ReliefE [124].

Our work on feature ranking has followed the same directions as our work on PCTs. We have extended Relief-based and tree-ensemble ranking methods to different output types (multi-target regression [106], MLC and HMLC[108]) and degrees of supervision, including fully supervised, semi-supervised [107] and unsupervised learning [105]. We have also addressed feature ranking for relational data [109] and for evolving data streams [97]. Relief-based methods estimate relevance directly from the data, others from the ensembles.

Global model explanations estimate how strongly the predictions of a model depend on its input variables. Some are specific to particular model classes, such as tree ensembles, and exploit model structure to calculate explanations efficiently. Model-agnostic approaches can be applied to any type of predictive model. Permutation importance [17] measures the decrease in performance after randomly permuting the values of a variable. Large decreases indicate the model relies strongly on that variable. Such estimates need careful interpretation when variables are correlated.

SHAP [87] explains model predictions by assigning a contribution to each variable for each individual prediction. The contributions, based on Shapley values from cooperative game theory, indicate how each variable shifts the prediction relative to a baseline. SHAP thus provides local explanations. Aggregating absolute SHAP values over examples yields a global ranking of variable importance.

LIME is a model-agnostic method for explaining individual predictions [114]. It perturbs the example being explained and fits a simple interpretable model to the resulting black-box predictions. The coefficients of this local approximation indicate which variables support or oppose the prediction. Since the explanation depends on how the neighbourhood is defined and sampled, it may vary between runs.

For image models, local explanations are often presented as saliency maps. These highlight the pixels or image regions that most strongly influence a prediction. Gradient-based methods and approaches such as Grad-CAM [120] are widely used for this purpose. They provide an indication of where the model focuses, but highlighted regions should not automatically be interpreted as causal explanations.

3 Foundation Models for Science

Machine learning traditionally starts from a specific task: a dataset is assembled and a separate model is learned from the available examples. This approach is often poorly matched to science. Scientific data can be costly to acquire, while their annotation may require specialised expertise, expensive instruments, or time-consuming experiments. Consequently, only a small number of labelled examples may be available, even when much larger collections of related unlabelled data exist.

Foundation models [14] provide a complementary approach. They are models pretrained at scale on broad collections of data and subsequently used or adapted for different downstream tasks. Instead of learning every task from the beginning, researchers transfer representations acquired during pretraining to a particular scientific problem. The best-known examples of foundation models (FMs) are large language models (LLMs), but FMs are also being developed for images, molecular sequences and structures, Earth-observation data, time series, and other scientific modalities. Their significance for science lies not simply in their size, but in their potential to transfer broadly learned representations to problems with limited task-specific data.

3.1 From Self-Supervised Learning to Foundation Models

Several learning paradigms address the availability of labelled data. In fully supervised learning, every training example is labelled with a target value. Semi-supervised learning combines a small labelled dataset with a larger set of unlabelled examples to learn a predictive model for a specified target task. Unsupervised learning uses no target variables and instead seeks regularities such as clusters or latent representations.

Self-supervised learning is a form of unsupervised learning: it starts from unlabelled data but constructs its own supervision [85]. Part of an observation is hidden or transformed to define a pretext task whose target can be generated automatically. The pretext task is not the ultimate objective. Its purpose is to make the model learn a representation that captures regularities in the data and can support subsequent tasks.

Self-supervised objectives take several forms. In masked language modelling, exemplified by BERT, selected tokens are hidden and predicted from their surrounding context [30]. Autoregressive language models instead learn to predict the next token from the tokens that precede it [21]. Masked modelling can also be applied to images: parts of an image are removed, and the model learns to reconstruct their content from the visible regions. This objective, used, for example, in masked autoencoders [52], is conceptually closer to masked language modelling than to next-token prediction.

In fact, autoencoders provide an early example of learning representations through reconstruction [53]. An autoencoder is trained to reproduce its input while passing the information through a constrained internal representation. To reconstruct the observation, the encoder must preserve salient aspects of its structure in this low-dimensional representation, which can subsequently be used as features for other tasks. Modern self-supervised methods use more elaborate architectures and objectives but follow the same general principle: useful representations can be learned before labels for the final task are introduced.

Self-supervised learning is closely connected to transfer learning [101]. In transfer learning, a model pretrained on a large source dataset is reused as the starting point for a related target problem. A visual model trained on a large general image collection can, for example, be adapted to a much smaller collection of medical or microscopy images. Its earlier layers may represent broadly useful visual patterns, while its later layers are replaced or adjusted for the new task [155].

The combination of self-supervised pretraining and transfer learning provides the basis for foundation models [14]. A high-capacity model is first trained on large quantities of data, commonly using a self-supervised objective. The resulting model provides representations and capabilities that can support many downstream tasks. A visual model pretrained through masked-image reconstruction may subsequently be used for classification, detection, or segmentation. Language models pretrained through masked-token prediction can support tasks such as classification, information extraction, and question answering, while autoregressive next-token models additionally provide a natural basis for text generation, giving generative AI its name.

A pretrained model can be used at different levels of adaptation [14]. In zero-shot use, it is applied to a task without task-specific training examples. For LLMs, one or several examples can instead be supplied as part of the input, enabling one-shot or few-shot in-context learning without changing the model parameters [21]. The model uses these examples as demonstrations from which to infer the task and the desired form of the output within the current interaction. Alternatively, the parameters of a pretrained model can be fine-tuned using a task-specific dataset. Adaptation thus ranges from direct reuse and in-context learning to extensive additional training.

This differs from conventional task-specific modelling, in which every problem requires its own training data and learning process. With foundation models, the expensive pretraining stage is shared across tasks. In this sense, foundation models can be regarded as a form of AI infrastructure for science: they provide reusable representations and capabilities that can be adapted to many scientific problems rather than learned anew for every task. This is particularly attractive in science, where large collections of unlabelled measurements may exist but labelled examples are scarce. Nevertheless, transfer is useful only when the pretrained representation captures information relevant to the target domain. Scientific usefulness must therefore be established empirically rather than inferred from model size.

3.2 Large Language Models and their Adaptation

LLMs are foundation models trained on very large text collections. Text is divided into tokens, which may represent words, parts of words, punctuation marks, or other character sequences. Tokens are converted into numerical representations and processed using the transformer architecture [145]. Its central mechanism, attention, relates each token to other tokens in the sequence, allowing the model to capture long-range dependencies and construct context-dependent representations of language.

As discussed in Section 3.1, language-model pretraining can use different self-supervised objectives, including the reconstruction of masked tokens from surrounding context [30]. The predominant autoregressive models instead predict each successive

token from the preceding sequence. At sufficient scale, this objective produces models capable of generating extended text and performing tasks not explicitly defined during pretraining [21]. The model generates a response by repeatedly estimating a probability distribution over possible next tokens, based on regularities learned from its training corpus and context provided in the current interaction.

A pretrained language model is not necessarily able to follow user instructions effectively. Its original objective is token prediction, whereas an assistant must interpret requests and produce relevant and appropriately structured responses. Instruction tuning addresses this difference by further training the model on examples of instructions and desired responses. Human preferences can additionally be used to encourage helpful behaviour and discourage unsuitable outputs [100]. The resulting model retains the broad capabilities acquired during pretraining but can apply them more effectively to tasks expressed in natural language.

The simplest form of adaptation is prompting. A prompt can specify the task, provide relevant context and examples, and define the desired output format. Prompting does not modify the model parameters and requires little or no task-specific training data. Its effectiveness, however, depends on the model already possessing the required knowledge and capabilities. Results may also be sensitive to prompt wording, the selection and order of examples, and the amount of information in the context window.

Fine-tuning modifies the pretrained model using data from the target domain or task. Continued pretraining on scientific publications can familiarise it with specialised terminology and discourse, while supervised fine-tuning can teach a particular task, such as extracting relations from articles or classifying clinical reports. Full fine-tuning updates all model parameters and is computationally demanding. Parameter-efficient methods instead modify only a small part of the model. Low-rank adaptation (LoRA), for example, learns low-rank modifications to selected layers while leaving the original parameters unchanged [55], reducing computational and storage cost.

Retrieval-augmented generation (RAG) provides another form of adaptation [81]. Before answering, the system retrieves relevant passages from an external collection, such as publications, experimental protocols, or curated scientific records, and includes them in the model input. Retrieval allows knowledge sources to be updated without retraining and can improve traceability by linking answers to source documents. It does not guarantee correctness, however: relevant evidence may not be retrieved, or the model may misinterpret it or generate unsupported claims.

The appropriate approach to adapting an LLM depends on the intended scientific use. Prompting is suitable when the LLM already has the required capability. Retrieval is valuable when answers must rely on a defined and changing body of knowledge. Fine-tuning is preferable when the model must learn specialised terminology, recurring task structures, or output conventions. These approaches can also be combined.

Our recent work provides a concrete example of domain adaptation in food and nutrition science. FoodyLLM was developed by fine-tuning Meta-Llama-3-8B-Instruct on approximately 225,000 task-specific question–answer pairs [46]. Low-rank adaptation was used to specialise the model while keeping the original model parameters fixed. The training data covered three groups of tasks: estimating the nutrient composition of recipes; assigning traffic-light labels for fat, saturated fat, sugar, and salt; and

recognising food entities and linking them to concepts from the FoodOn, SNOMED CT, and Hansard ontologies. As an example for the first, FoodyLLM might receive a prompt "*Determine the nutritional profile per 100 g of a recipe containing 180 g bread flour, 180 g cake flour, 200 ml sugar, 200 g butter, 200 ml water, and a pinch of salt*" and respond "*Energy: 364.03 kcal; fat: 17.96 g; protein: 4.09 g; salt: 0.05 g; saturated fat: 10.95 g; sugars: 18.33 g*". Overall, FoodyLLM combines quantitative assessment and classification with the semantic standardisation needed to make heterogeneous food data interoperable.

FoodyLLM was compared with the original Llama 3 8B model and Gemini 2.0 using zero-, one-, and five-shot prompting. It substantially outperformed both general-purpose models across the three task groups, while providing additional examples in the prompt did not close the performance gap. The results demonstrate that in-context examples cannot necessarily substitute for domain-specific fine-tuning when a task requires specialised quantitative and semantic knowledge. At the same time, the study illustrates how the different adaptation mechanisms discussed above complement one another: a broadly pretrained model was specialised efficiently through LoRA and could subsequently be used with zero-shot or few-shot prompts for several related tasks.

There are many limitations to the use of LLMs in science. LLMs can generate plausible but incorrect statements, reproduce biases in their training data, and provide unstable answers to equivalent questions [157]. They should therefore not be treated as repositories of verified scientific knowledge. Their value is greatest when generation is grounded in traceable sources, combined with structured data and computational tools, and subjected to domain-specific evaluation. In this role, LLMs act as flexible interfaces to scientific information, not as substitutes for scientific expertise.

3.3 Non-textual and Multimodal Foundation Models

Foundation models are not limited to language. Scientific data also take the form of images, biological sequences, molecular graphs and three-dimensional structures, spectra, time series, spatial fields, and measurements produced by scientific instruments. Models pretrained on large collections of such data can acquire representations that are subsequently reused across tasks. The principle is the same as in Section 3.1: self-supervised learning extracts regularities from unlabelled data, reducing the amount of labelled data required for a specific application. The pretraining objective and model architecture must, however, reflect the structure of the modality. Image representations can be learned by trying to reconstruct masked regions, while models of sequences, graphs, or spatial structures must preserve their characteristic relationships/ constraints.

Visual foundation models provide a prominent example. The Segment Anything Model (SAM) was trained on 11 million images and more than one billion segmentation masks to identify objects from prompts such as points or bounding boxes [62]. We used its successor, SAM2, for segmenting electron-microscopy images of barium hexaferrite nanoplatelets [88]. Our objective was to identify individual particles and derive their diameter distributions, which are important for controlling the magnetic properties and colloidal stability of the material.

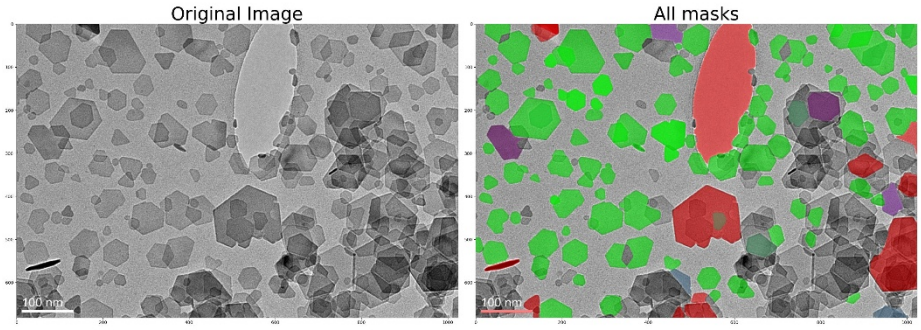

Fig. 3. Example segmentation of nanoplatelets. The original image is on the left, the right image shows clustering results (selected best cluster is green, outliers are red).

Because labelled electron-microscopy data are scarce and expensive to produce, SAM2 was applied without task-specific training. Although it detected most relevant particles, it also segmented other structures in the images. We therefore clustered the masks produced by SAM2 according to their colour, texture, shape, and size, allowing the nanoplatelets to be distinguished from irrelevant structures (see Fig. 3). Evaluation on 185 images from 15 electron-microscopy analyses showed that clustering improved the agreement between the automatically and manually derived particle-diameter distributions.

This illustrates an attractive use of foundation models in science: knowledge acquired from large external datasets can be transferred to specialised problems for which annotations are scarce, while lightweight unsupervised methods can adapt the resulting outputs without fine-tuning the model. As a model trained predominantly on ordinary images does not automatically recognise which structures are scientifically relevant, its outputs must be evaluated against domain-specific criteria and interpreted by scientists.

Foundation models have also been developed for other scientific modalities. In biology, models learn from genomic and protein sequences, while multimodal foundation models seek to integrate genomics, transcriptomics, epigenomics, proteomics, metabolomics, and spatial profiling [28]. In chemistry and materials science, they can combine symbolic, graph-based, geometric, and textual representations [136]. Preserving such native representations is important: converting structures or measurements into text provides a convenient interface but may discard quantitative, spatial, or relational information.

Many scientific questions require several modalities to be considered together. Earth observation is a clear example: satellites observe the same region using different instruments, spectral bands, spatial resolutions, and acquisition times. Multimodal foundation models can jointly process these complementary observations, allowing information from one modality to support another. TerraMind [58], a generative AI system for Earth observation, was pretrained on nine geospatial modalities and can process or generate different modality combinations within a single model. DOFA uses wavelength information to adapt a shared model dynamically to data from different sensors [154]. As many Earth foundation models emerge, a recent synthesis identifies eleven desirable

characteristics for them, including multisensor integration, wavelength and scale awareness, and physical consistency [158].

Scientific multimodality is therefore broader than the familiar combination of text, images, audio, and video. In inorganic chemistry, for example, evidence is distributed across publications, chemical formulae, molecular and crystal structures, spectra, numerical measurements, crystallographic databases, simulations, and experimental records [64]. Integrating these sources requires both learned representations and models that respect chemical and physical constraints.

There are two broad ways to achieve such integration. Several modalities can be incorporated into a single foundation model, enabling it to learn relations among them directly. Alternatively, specialised models can retain their native representations and be connected within a larger system, with an LLM providing a natural-language interface and coordinating the different models, not replacing their specialised capabilities.

3.4 From Foundation Models to Agents in Science

The integration strategy based on LLM orchestration leads from individual foundation models to agentic systems that combine models with external knowledge, computational tools, and experiments in pursuit of scientific goals. Whereas an FM produces an output in response to an input, an agent can formulate intermediate objectives, select and use tools, evaluate results, and revise its actions. An LLM often provides the natural-language interface and coordinates this process, but the system may include specialised models, databases, simulators, code interpreters, and laboratory equipment.

A scientific agent typically operates through an iterative loop. It interprets a goal, develops a plan, performs one or more actions, observes their results, and decides what to do next. Its actions might include searching the literature, retrieving data, running an analysis, calling a predictive model, executing a simulation, or proposing an experiment. Information returned by these tools becomes part of the agent's working context and can alter its subsequent plan. This ability to act on intermediate results distinguishes agents from fixed workflows, although many practical systems constrain their actions through predefined tools, protocols, and decision points, a constraint already visible in the robot scientists Adam, Eve, and Genesis (Section 5.2), which anticipated this loop and formalised it through an explicit ontology rather than an LLM-based coordinator.

Chemistry provides prominent early examples. ChemCrow [15] connects an LLM to expert-designed tools for tasks including molecular calculations, reaction planning, and safety assessment. Coscientist [13] goes further by combining literature and web search, code execution, documentation for laboratory equipment, and robotic experimentation. Given a high-level objective, it planned chemical syntheses and executed some of them using automated laboratory hardware. In computational research, the AI Scientist [86] generated research ideas, wrote and modified code, ran experiments, analysed results, and produced papers in selected areas of machine learning. Multi-agent systems can additionally assign specialised roles to different agents. The Virtual Lab, for example, organised agents representing different scientific disciplines to design nanobody candidates, some of which were subsequently validated experimentally [132].

These systems illustrate different degrees of agency and autonomy. Agency concerns the ability to pursue a goal through tool-mediated actions and feedback; autonomy concerns how much of this process occurs without human intervention. A system may therefore be agentic while scientists still define the objective, approve plans, select among hypotheses, or authorise experiments. Such human oversight is particularly important when actions are costly, irreversible, or potentially unsafe.

Scientific agents inherit the limitations of their component models and introduce new ones. Errors can propagate across steps, tools may be misused, and a coherent-looking plan may rest on false premises: mistaking predictive fluency for validity (cf. Section 2.1) is even costlier once a model's outputs feed its own next action. Reliable use requires explicit constraints, validated tools, records of data and actions, and evaluation of intermediate/ final results. The most promising role of agents is not to replace scientists, but to connect knowledge, models, and experimentation into traceable workflows in which routine steps are automated and decisions remain open to scientific scrutiny.

4 Automated Scientific Modelling

Models are central to science. They provide formal representations of systems, summarize current understanding, generate predictions, and support reasoning about situations not yet observed. Models of dynamical systems are commonly expressed as ordinary differential equations (ODEs), which describe how system states change over time. Scientists use them to simulate behaviour under different conditions, examine sensitivity to changes, and investigate the mechanisms responsible for observed phenomena.

Scientific models are traditionally constructed by domain experts. This knowledge-driven approach incorporates established theories, plausible mechanisms, and detailed understanding of the system under study. However, constructing, calibrating, and validating models is knowledge-intensive, time-consuming, and expensive. As scientific data have become more abundant, researchers have increasingly turned to data-driven methods that construct models automatically from observations.

Automated modelling can accelerate science, but a model that accurately predicts an observed trajectory is not necessarily scientifically meaningful. Its variables and mathematical terms may not correspond to recognizable system components and processes, while its behaviour under new conditions may be implausible. Domain scientists may therefore be unable to relate it to their existing knowledge.

The objective of automated scientific modelling is not merely to find mathematical expressions that fit data, but to produce models whose structure, assumptions, and mechanisms can be understood, examined, and reused. This section traces the development from equation discovery and symbolic regression to knowledge-guided equation discovery and process-based modelling, concluding with approaches that learn generative models or model priors from sets of equations and broader scientific knowledge.

4.1 From Equation Discovery to Symbolic Regression

Conventional regression estimates the parameters of a predefined model structure. For example, linear regression assumes a relation of the form $y=ax+b$ and determines suitable values of a and b from a set of measurements (data) about x and y. Equation discovery addresses the more difficult problem of determining both the structure of the equation and its numerical parameters. The target relation may be linear, polynomial, rational, exponential, or composed of several mathematical functions. The space of possible structures is therefore discrete, combinatorial, and effectively unlimited.

The computational study of equation discovery began within artificial intelligence in the late 1970s. Langley's BACON system used heuristic search to discover regularities among measured variables [71]. It examined whether variables remained constant or changed together and constructed new quantities through operations such as multiplication and division. Applying such heuristics recursively allowed BACON and its successors to rediscover forms of scientific laws associated with Kepler, Ohm, Coulomb, and the ideal gas. This work contributed to a broader programme of computational scientific discovery in which regularities, concepts, and explanatory hypotheses could be generated through heuristic search [76].

The term *symbolic regression* became prominent through Koza's work on genetic programming (GP) in the early 1990s [69]. In GP, candidate equations are represented as expression trees and evolved using operators inspired by biological evolution. Selection favours expressions that fit the observations, crossover exchanges subtrees between candidates, and mutation alters parts of an expression. The method thus searches simultaneously over mathematical structures and their parameter values.

Equation discovery and symbolic regression address essentially the same computational task, but the terms reflect different research traditions. Equation discovery originated in artificial intelligence and emphasized scientific laws, background knowledge, and interpretation. Symbolic regression developed mainly within GP and ML, emphasizing general-purpose search through spaces of mathematical expressions.

Modern approaches increasingly combine ideas from both traditions. They balance fit against complexity as highly complex expressions may reproduce noise, whereas simpler equations are easier to interpret and more likely to generalize. Several chapters in this volume examine contemporary approaches to symbolic regression in greater depth, including scientifically guided experimentation, deep symbolic optimization, neural-guided search, and generative models for equation discovery [22, 51, 91, 93].

Equation discovery can also be applied to dynamical systems. An ODE model represents the rate of change of a system variable x_i as a function of the current system state and constant parameters: $dx_i/dt = f_i(x,\theta)$. When time-series observations are available, their numerical derivatives can be estimated and treated as target variables. Equation discovery can then search for the functions f_i. LAGRANGE, introduced in 1993, used this approach to discover polynomial ODEs [39]. It estimated the derivatives of observed variables and searched for polynomial relations between these derivatives and the state variables. Later methods adopted related formulations. SINDy, for example, discovers sparse equations by selecting a small number of terms from a predefined

library of candidate functions [23], while other methods search more general nonlinear expressions [119].

Our application of GoldHorn, a successor to LAGRANGE designed to deal with noisy data, to modelling algal growth in the Venice Lagoon exposed a fundamental limitation of purely data-driven equation discovery [65]. The discovered equations fitted the observations, but aquatic ecologists did not find them meaningful. Their mathematical terms did not correspond to recognizable ecological processes and could not easily be related to existing knowledge.

An equation is therefore not necessarily explanatory merely because it is expressed symbolically. In a manually constructed ecological model, variables represent identifiable system components, such as phytoplankton and zooplankton populations, while terms represent processes such as growth, mortality, nutrient uptake, and grazing. This process context is normally explained in the publication accompanying the model but is not contained in the equations themselves. Once separated from that context, even a concise ODE may be difficult to understand.

The Venice Lagoon experience consequently shifted the central question: from how to discover equations that fit observations to how to discover equations that also reflect the concepts and mechanisms through which domain scientists understand the system.

4.2 Integrating Data and Domain Knowledge

Knowledge-driven and data-driven modelling are two ends of a spectrum. In knowledge-driven modelling, domain experts fix a model's components, processes, and structure, and data mainly estimate parameters and evaluate behaviour. In data-driven modelling, structure is inferred primarily from data, with domain knowledge entering only indirectly, through the choice of variables, operators, or evaluation criteria. Knowledge-driven models support scientific explanation but are expensive to build and may inherit established assumptions; data-driven methods explore many alternative structures but may produce models that violate known principles or resist interpretation. Automated scientific modelling should combine the two, rather than choose one.

Every equation-discovery method embodies some inductive bias: no equation discovery system searches the space of all possible mathematical expressions. Its representation fixes which variables, operators, and structures can be considered; its search method favours some regions of the space over others; its complexity measure encodes assumptions about which models are preferable. Making these biases explicit allows them to be examined and adapted to the domain.

Context-free grammars offer a declarative way to define the language of admissible equations, through production rules specifying how expressions can be built. A general grammar might allow arithmetic operators and trigonometric functions, while a domain-specific one may restrict the search to structures matching known processes. Our first attempt to incorporate domain knowledge into equation discovery allowed for subexpression templates to be specified in the context of genetic programming [38].

LAGRAMGE subsequently separated the representation of domain knowledge from the search procedure entirely [139]. A context-free grammar defines the space of admissible ODEs, heuristic search explores it, and numerical optimization fits each

candidate's parameters. This separation mattered because the same declarative model space could, in principle, be searched by different algorithms.

Grammar-based discovery improved both efficiency and scientific relevance. Mathematical knowledge excluded malformed expressions; dimensional or structural restrictions ruled out combinations inconsistent with the domain; and, more importantly, grammar symbols could refer to scientifically meaningful processes rather than bare mathematical operations. Applying this approach to phytoplankton growth in Lake Glumsø, Denmark [138] resulted in models that were received more positively by ecologists than the earlier Venice Lagoon equations, precisely because their terms mapped onto recognizable ecological processes.

This success exposed a further obstacle. Domain experts could interpret and evaluate the process-oriented models, but could not themselves express their knowledge as context-free grammars — a knowledge engineer was still needed to elicit ecological knowledge and translate it into grammar rules. Grammars were an effective machine representation of model constraints, but not a natural language for communicating with domain experts. What was needed was a representation preserving the computational precision of grammars while exposing model components in the vocabulary scientists already use.

4.3 From Equations to Process-based Models

Process-based modelling represents dynamical systems explicitly in terms of their components and interactions [20, 40, 73]. Instead of presenting domain scientists with a grammar or an unannotated system of equations, it describes a model through two kinds of components:

- *entities*, representing parts of the system, such as organisms, populations, substances, or compartments;
- *processes*, representing interactions or transformations, such as growth, predation, diffusion, reaction, or decay.

Each process has a mathematical interpretation, which contributes one or more terms to the differential equations describing the participating entities. A process-based model (PBM) has two connected representations: a higher-level representation in terms familiar to domain scientists, and a lower-level representation with the corresponding equations required for parameter estimation and simulation. The representation and computational discovery of PBMs are examined in depth by Langley [72] (this volume).

For example, an aquatic-ecosystem model may contain phytoplankton and zooplankton populations as entities. A grazing process links them and contributes a loss term to the equation for phytoplankton and a growth term to the equation for zooplankton. The terms are understandable because their origin and role are explicit: they represent a hypothesized interaction rather than merely components of a fitted expression.

Simulation takes place at the level of equations, but the resulting behaviour can be explained at the higher level by referring to the processes responsible for it. A decline in phytoplankton may be attributed to grazing, nutrient limitation, mortality, or a combination of these processes. PBMs can thus support causal reasoning, with strength of the causal claims depending on the underlying assumptions/ evidence [75].

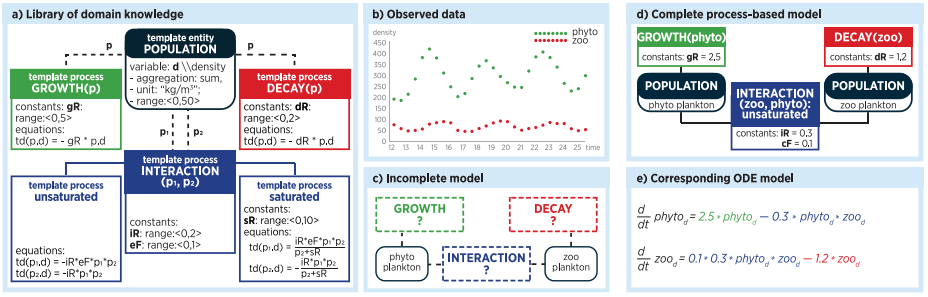

Fig. 4. Process-based modelling of dynamical systems. Domain knowledge (a), observed data (b), and structural constraints (c) are combined to construct a complete model expressed at two connected levels: scientifically meaningful entities and processes (d), and the corresponding system of ordinary differential equations (e).

Figure 4 illustrates the PB modelling task. Domain knowledge specifies the possible entities and processes, commonly organized in a library of reusable model components. Observations describe the system's behaviour, while model constraints identify components known to be present or restrict the alternatives to be considered. Automated modelling searches for a complete model consistent with these inputs. It selects and instantiates suitable entities and processes, generates the corresponding equations, estimates their parameters, simulates the model, and evaluates its agreement with the observations. The output is not merely a fitted system of equations, but a structured scientific model whose assumptions can be inspected and related to domain knowledge.

A process library captures general knowledge about a domain, while an individual model instantiates an appropriate subset for a particular system. In aquatic ecology, an environmental engineer developed an extensive library for modelling population dynamics [8]. Combined with system-specific observations and constraints, it was reused to model ecosystems ranging from the Ross Sea in Antarctica [5] to Lakes Bled in Slovenia [6] and Kasumigaura in Japan [7]. Models need not be constructed independently for every ecosystem, nor must all ecosystems be assumed to have identical structures. Reusable knowledge and system-specific evidence play complementary roles.

Our initial approach encoded knowledge about processes in population dynamics and transformed it into grammars, then invoked LAGRAMGE, although the term *process-based models* was not yet used [40]. A later implementation integrated process-based knowledge and model constraints within equation-discovery [140]. ProBMoT searched the space of process-based models directly and exhaustively [26] and was later extended with *compartments* to model spatially or structurally distributed systems. Compartments can contain entities and processes, as well as nested sub-compartments, representing complex systems hierarchically. In catchment modelling, they represent sub-catchments with different land-uses and process formulations (cf. Radinja et al. [113], this volume).

Process-based modelling was also extended beyond ecology to model cellular immune response, synthetic biological circuits, and epidemic outbreaks by Tanevski et al. [133–135]. These applications introduced further challenges: model quality may involve

several competing criteria; systems may have to be designed to produce desired behaviour even without an observed trajectory; and some phenomena require stochastic rather than deterministic models.

The process-based modelling approach does not eliminate the difficulty of scientific discovery. Its results depend on the quality of the knowledge library, the informativeness of the observations, and the constraints imposed on the search. If an important process is absent from the library, no model containing it can be discovered; if the library is too permissive, the model space may become prohibitively large.

The central advantage is therefore not that process-based models are automatically correct, but that their hypotheses are explicit and open to examination. Scientists can challenge and revise the selected entities, processes, equations, parameters, and assumptions. The model becomes a medium for interaction between computational search and scientific expertise rather than a result delivered by an opaque algorithm.

4.4 Learning Generative Models of Equations

The search space of symbolic regression, consisting of equation structures, can itself be described by a generative model. Such a model specifies how candidate equations are constructed and, in probabilistic variants, how likely different structures are. It thereby provides a prior over equations: candidates considered plausible receive greater attention, while implausible ones are generated less frequently or excluded.

Grammars are a transparent form of generative model. Their production rules define how variables, constants, operators, and functions can be combined into valid expressions. The grammar may be generic or encode domain knowledge, such as dimensional constraints and plausible process combinations. Probabilistic context-free grammars associate probabilities with the production rules. They can favour simple expressions commonly used in a scientific domain without excluding less likely alternatives.

ProGED, introduced by Brence et al. [18], uses probabilistic grammars to generate candidate equation structures, fits their numerical constants, and evaluates the resulting equations against observations. The grammar thus separates prior knowledge about plausible structures from evidence supplied by the data. However, specifying suitable rules and probabilities manually can be knowledge-intensive. An alternative is to learn the rule probabilities from a corpus of existing equations, thereby capturing recurring patterns in how variables, functions, and operators are combined [27].

Neural generative models provide a less explicit approach. Variational autoencoders (VAEs) encode equations, as token sequences or expression trees, into a continuous latent space and decode points back into symbolic expressions, so search can partly occur in a learned space where related equations lie close together. Hierarchical VAEs (HVAEs) additionally represent equations at several levels, reflecting their nested structure. EDHiE combines hierarchical representations of equations with evolutionary search, using the learned representation to guide the generation of candidate structures [90]. Grammar-based and neural generative models, including ProGED and EDHiE, are treated in greater depth by Mežnar et al. [91] in this volume.

Learning a generative model from published equations resembles pretraining a foundation model: regularities from prior examples are reused to solve new problems,

making search more efficient and directing it toward familiar structures. This prior may also reproduce the corpus's conventions and blind spots — unconventional but valid equations may receive low probability, particularly from a narrow corpus.

Large language models offer a further form of learned generative model, acquiring regularities from text, mathematical expressions, and code rather than only a curated equation collection. Recent approaches use LLMs to propose and refine equation structures, often as programs, while external optimizers estimate constants and observations to determine equation quality [89, 121]. LLMs can also translate natural-language knowledge or requirements into constraints on symbolic regression [83], or evaluate candidates for dimensional consistency, simplicity, and plausibility [137].

These approaches extend symbolic regression from searching a predefined space to exploiting knowledge learned from prior equations and scientific discourse. Yet their output remains hypothetical: benchmark performance may partly reflect memorization, while convincing expressions may be scientifically invalid. Learned generative models should guide exploration, not replace fitting to observations, testing under new conditions, and evaluation against domain knowledge.

5 Semantic Technologies for Open Science

Scientific knowledge must be communicated precisely if it is to be verified, reproduced, combined with other knowledge, and reused. This is especially important for computational scientific discovery approaches, such as the ones discussed in Section 4, which depend not only on methods for learning from data, but also on explicit representations of domain knowledge, scientific models, hypotheses, constraints, and evidence.

While natural language remains indispensable for scientific communication, it also leaves room for ambiguity and unstated assumptions. The same term may have different meanings in different disciplines, while different terms may denote essentially the same concept. Important information about experimental conditions, data preparation, parameter settings, or evaluation procedures may be distributed across an article, supplementary material, software documentation, and data repositories. Such variation is manageable for an informed human reader, but it presents a fundamental obstacle to computational processing. Formalization complements conventional scientific communication by making the entities involved in research, their properties, and their relationships explicit and machine-interpretable.

5.1 Why formalize science?

Formalization serves three closely related purposes. The first is precision. A formal representation can specify exactly which dataset was analysed, which task was addressed, which algorithm and implementation were used, how their parameters were set, which experimental protocol was followed, and how the resulting model was evaluated. It can distinguish concepts that are easily conflated in ordinary language, e.g., an algorithm as an abstract procedure, its implementation in a particular software package, and the execution of that implementation with specific inputs and parameter values.

The second purpose is automation. Computers can execute scientific procedures only if the relevant objects, operations, goals, and constraints are represented in a form they can process. Laboratory automation may perform a predefined sequence of operations, but more autonomous systems must also select experiments, interpret observations, revise models or hypotheses, and decide what to do next. This requires explicit descriptions not only of data and materials, but also of experimental actions, the inputs they require, the outputs they produce, and the conditions under which they are applicable.

The third purpose is sharing and open science. Open science treats scientific knowledge as a public good that should be accessible and reusable rather than confined within publications, proprietary systems, or institutional silos. Open access to articles is only one part of this objective. Data should follow the FAIR principles: they should be findable, accessible, interoperable, and reusable [152]. The same principles increasingly apply to software, models, workflows, experimental protocols, and results. Merely placing these research objects in a repository, however, does not make them interoperable or reusable. They must be accompanied by sufficiently precise metadata, expressed through shared concepts and vocabularies, for other researchers and computational systems to interpret them correctly.

Semantic technologies provide methods for constructing such descriptions. Their central component is the ontology, commonly defined as a formal, explicit specification of a shared conceptualization [48]. An ontology identifies the principal concepts or classes in a domain, the relations among them, and, where appropriate, individual instances. As these components are represented in formal languages, ontologies can support semantic search, integration of heterogeneous information, consistency checking, automated reasoning, and reuse across information systems.

Ontologies can be developed at different levels of generality. Foundational or upper-level ontologies, such as BFO, the Basic Formal Ontology [4], describe general categories such as objects, processes, qualities, roles, and information entities. Domain ontologies represent more specific knowledge about areas such as biology, chemistry, materials science, scientific experimentation, machine learning, or optimization. Alignment with upper-level ontologies can provide a common conceptual basis for connecting knowledge developed independently in different scientific domains. A more detailed account of knowledge engineering for automated scientific discovery, including ontologies for experiments, experimental actions, hypotheses, computational investigations, and robot scientists, is provided by Soldatova [127] (this volume).

The biological sciences were among the earliest adopters of infrastructures combining repositories, standardized metadata, and shared ontologies. The Gene Expression Omnibus, for example, was established as a public repository for heterogeneous data from high-throughput gene-expression and genomic-hybridization experiments [41]. The OBO Foundry coordinates the development of interoperable biomedical ontologies, seeking to reduce conceptual fragmentation and support data integration across biological and medical domains [126]. Scientific models are also increasingly treated as reusable research objects. BioModels preserves computational models together with structured information needed to interpret, reproduce, and reuse them [82]. These examples show how the FAIR perspective can be extended from observational data to formal representations of scientific knowledge.

Formalization is therefore central to open science in a stronger sense than simply making research outputs publicly available. It allows those outputs to become part of an interconnected body of machine-actionable scientific knowledge.

5.2 Ontologies and Automated Science

The importance of formal descriptions becomes particularly apparent in robot science. King et al. [60] defined a robot scientist as a physically implemented laboratory automation system that uses AI to perform cycles of scientific experimentation. Such a system generates hypotheses, designs experiments to test them, physically executes the experiments, interprets the results, and repeats the cycle. Gower et al. [47] (this volume) examine the philosophical and methodological foundations of such systems in greater depth and interpret the scientific discovery cycle as a form of active learning, in which the system selects experiments expected to improve its scientific model.

The robot scientist Adam demonstrated this architecture in functional genomics. It formulated hypotheses about genes encoding orphan enzymes in yeast, designed experiments to test them, executed the experiments using laboratory robotics, and interpreted the resulting measurements. Adam's investigations were formally described using LABORS, an ontology derived from the EXPO ontology of scientific experiments [128], connecting the hypotheses, experimental designs, procedures, and metadata from which its results arose [61]; the development of the corresponding knowledge representations across Adam, Eve, and Genesis is examined comparatively by Soldatova [127]. The formal description was not an addition produced after the experiments had been completed. It was a consequence of designing the system so that its scientific reasoning and experimental actions were represented explicitly, and the resulting provenance record supported interpretation, reproducibility, and subsequent reuse of the experiments.

Eve extended the robot-scientist approach to drug discovery. It combined hypothesis generation and compound selection with automated screening, modelling, and experimental validation [153]. Adam and Eve illustrate a general principle: increasing the autonomy of a scientific system increases, rather than reduces, the need for precise representation. The system must know what constitutes a hypothesis, which experiment can test it, how an experimental outcome relates to the hypothesis, and how the new evidence should affect subsequent decisions.

The next generation of this work is represented by Genesis, a robot scientist being developed for systems biology around a microfluidic platform containing 1,000 computer-controlled microbioreactors. Genesis is intended to iteratively improve a systems-biology model of yeast by generating hypotheses, selecting and conducting experiments, comparing predicted and observed phenotypes, and revising the underlying theory. Its LGEM+ modelling framework represents metabolic knowledge in first-order logic and uses automated deduction and abduction to produce testable hypotheses. Controlled vocabularies provide the semantic correspondence between computational models, experimental inputs, and measured outputs [47].

The same architecture has increasingly been adopted in materials science, usually under the terms self-driving laboratory or materials acceleration platform (MAP). These systems combine machine-learning or model-based experiment selection with

automated synthesis and characterization in a closed feedback loop. Materials science is particularly suitable for this approach because potentially useful materials inhabit vast combinatorial spaces, while their properties often depend on complex interactions among composition, synthesis conditions, structure, and processing.

An influential demonstration involves a fully autonomous mobile robotic chemist [25]. Rather than automating a single instrument or fixed experimental line, the robot moved through a conventional laboratory and operated equipment designed for human researchers. Guided by Bayesian optimization, it performed 688 experiments over eight days in a ten-variable search space and identified photocatalyst mixtures for hydrogen production that were six times more active than the initial formulations. The approach was described as "automating the researcher rather than the instruments," emphasizing its potential to connect heterogeneous laboratory equipment through a mobile robotic operator.

A complementary development is the distribution of an autonomous discovery process across multiple laboratories. The FINALES platform [149] coordinates computational and experimental capabilities contributed by different institutions to optimize battery-electrolyte formulations, connecting optimization algorithms, electrolyte preparation and characterization, cell assembly, testing, and lifetime prediction, with no single instrument or laboratory implementing the entire discovery cycle. This only works because the participating platforms share data schemas, vocabularies, and interfaces precise enough to preserve units, conditions, and provenance across institutional boundaries — the same requirement for formal representation that connected the reasoning and measurements of a single robot scientist like Adam, now operating at the scale of an entire distributed research infrastructure. We are developing similar facilities at the Jožef Stefan Institute, where we are also applying machine learning and multi-objective optimization for materials design: A smaller-scale example involving the design of foamed glass is discussed in Section 6.3.

5.3 Semantic Technologies for Empirical Computer Science

Our work on semantic technologies has focused particularly on the formal description of machine learning (ML)/ data mining, and optimization. These fields are themselves empirical sciences: new methods are evaluated through computational experiments involving tasks (ML tasks include datasets), algorithms, software implementations, evaluation procedures, and performance results. Although such experiments are easier to automate than physical experiments, they can be difficult to reproduce because essential information is absent, ambiguous, or represented differently by different platforms.

The OntoDM family of ontologies, developed by Panov et al. [103], provides a systematic representation of the principal entities involved in data mining and knowledge discovery. It is based on a framework in which a data-mining task specifies the type of generalization to be obtained from a given type of data, subject to appropriate constraints, while a data-mining algorithm specifies a procedure for solving such a task.

OntoDT [102] represents datatypes, which determine the structure of the objects that an algorithm consumes and produces. Tabular data, sequences, networks, images, and text require different representations and permit different operations, while the output

datatype helps distinguish tasks such as classification, regression, clustering, multi-target regression, multi-label classification, and hierarchical multi-label classification. OntoDM-core represents datasets, tasks, algorithms, implementations, and generalizations such as predictive models, patterns, and clusterings [103]. These are related but partly orthogonal dimensions: a task describes what is to be achieved; an algorithm specifies an abstract procedure; an implementation realizes it in software; and an execution applies that implementation to particular inputs under specified parameter settings. Separating these entities avoids treating an algorithm, a program, and one of its executions as interchangeable.

OntoDM-KDD extends this representation to complete knowledge-discovery processes. It describes actions, their inputs and outputs, and their organization within workflows and scientific investigations. OntoDM as a whole can represent that a particular implementation of an algorithm was applied to a specified dataset, using particular parameters and an experimental design, to address a given task and produce a model evaluated using specified measures. Its place in the broader landscape of ontologies for computational scientific investigations is discussed by Soldatova [127].

One important application of these representations is the construction of experiment databases. Scientific articles can report only a fraction of the results generated by extensive benchmarking studies. Experiment databases can record many algorithms applied to many datasets under different parameter settings and evaluation procedures, together with the metadata needed to reproduce and compare the runs [143]. This has led to OpenML, a platform for sharing datasets, machine-learning tasks, implementations, experimental runs, and results [144].

Systematically represented experiments can themselves become objects of scientific analysis. Researchers can investigate which algorithms perform well on which problems, how performance depends on dataset properties, and which parameter settings are effective under particular conditions. Such questions underpin meta-learning, where knowledge derived from previous experiments can recommend algorithms or configurations for new datasets and thus support automated machine learning.

Although AutoML commonly seeks to improve predictive performance through automated algorithm selection and configuration [56], meta-learning can also pursue an explanatory objective: understanding how dataset properties and algorithm design or configuration jointly influence performance. Bogatinovski et al. [12], for example, analysed the performance of many different multi-label classification methods across 40 datasets using 50 meta-features. They found properties of the label space (relationships among labels) particularly important for explaining performance.

Semantic descriptions also enable more precise retrieval than keyword-based search. Researchers might ask for algorithms applicable to hierarchical multi-label classification, experiments using a particular validation design, or models learned from graph-structured data and evaluated with a specified measure. Such questions can be expressed in SPARQL [103]. Large language models can translate natural-language questions into formal queries and present the results, while the ontology supplies the explicit concepts and relations against which the queries are evaluated.

The same approach applies to optimization, where the OPTION ontology represents algorithms, benchmark problems, experimental settings, evaluation measures and

results [68]. Benchmarking platforms often use incompatible formats and conceptual models. OPTION enables integration, comparison, querying, and reuse of results, supporting investigations of algorithm behaviour across problem classes.

Semantic technologies will become increasingly important as AI agents assume larger roles in science. Agents that discover datasets, select methods, invoke software, conduct experiments, and interpret results must understand available actions, their inputs and outputs, and applicable goals and constraints. Resources designed for humans may not support intensive autonomous use, which requires stable sources, rigorous schemas, explicit uncertainty, and continuous updating [127].

Such support is provided by ontologies, which can describe the environment, constrain inappropriate actions, and record decision and workflow provenance. LLMs provide natural-language interaction and broad background knowledge; ontologies provide explicit semantics, domain constraints, and links to validated scientific objects. Together, they connect open and automated science by organizing datasets, software, models, experiments, and results into an interpretable, reusable structure supporting human collaboration and increasingly autonomous computational/ laboratory systems.

6 Applications in the Sciences

The AI methods discussed in the preceding sections have been applied to a wide range of problems across the natural sciences. This section presents representative examples from three domains — life sciences, environmental sciences, and materials science — drawn primarily from work carried out by myself and colleagues at the Jožef Stefan Institute, in collaboration with an extensive network of domain science experts, both Slovenian and international. The selection is illustrative rather than exhaustive; the environmental and life sciences portfolios are particularly extensive.

6.1 Life Sciences

Applications of AI in the life sciences span medicine, molecular biology, and systems biology. In the biomedical domain, we have addressed medical diagnosis and prognosis, identification of biological signatures of disease, drug design and repurposing, and prediction of clinical outcomes. A recurring theme is the use of multi-target prediction methods to handle the inherent complexity of biomedical data, where multiple related outcomes must be predicted simultaneously from heterogeneous input features.

One well-developed application concerns survival analysis — the prediction of time until an event of interest such as death or disease progression. We reformulated this as a semi-supervised multi-target regression problem, treating censored observations as partially labelled examples rather than discarding them [116]. Each distinct event time becomes a separate regression target, with missing values assigned for all time points after censoring. Evaluated on eleven real-life clinical datasets, the approach performs comparably to or better than competing approaches in predictive accuracy and calibration, while producing smaller and interpretable models. Applied to a large ALS (Amyotrophic Lateral Sclerosis) clinical trials dataset with thousands of patients and

hundreds of features, an interpretable single-tree model recovered clinically meaningful predictors including forced vital capacity, ALS functional rating, age, and weight trajectory, with one-year survival probability ranging from 94% in the most favourable patient group to 30% in the most severe.

In drug design, we applied virtual compound screening to discover of host-directed antimicrobials against *Mycobacterium tuberculosis* and *Salmonella typhimurium* [67]. The approach builds predictive models from compound structure descriptors (functional groups, molecular fingerprints, bulk physicochemical properties), enriched with information on the protein targets of each compound retrieved from databases such as PubChem, including the functional annotations and pathway memberships of those targets. Models trained on a library of around 1,200 pharmacologically active compounds (LOPAC) were then applied to screen millions of compounds from PubChem, yielding ranked candidate lists for experimental follow-up. Multi-target prediction focused on bacterial load reduction and host cell toxicity — two properties that must be optimised simultaneously for a compound to be a viable therapeutic candidate. A similar approach was applied to identify anti-fibrotic compounds for pulmonary fibrosis: tree ensemble models trained on high-content screening data from 640 FDA-approved compounds were used to virtually screen the ChEMBL database, identifying dopamine and its novel derivative TS1 as potent inhibitors of myofibroblast differentiation, subsequently validated in murine models of lung fibrosis [115].

In molecular biology, we have addressed the task of genome-wide gene function prediction in bacteria, a problem of increasing urgency as the number of sequenced genomes grows rapidly while a large fraction of genes remain functionally unannotated — even in well-studied organisms such, as *Mycobacterium tuberculosis*, more than a quarter of genes were of unknown function at the time of our study. We used tree ensembles for hierarchical multi-label classification, with the Gene Ontology as the annotation scheme, initially over phyletic profiles describing the similarity of genes to genes in other organisms [125]. Follow-up work considered five different feature sets (including phyletic profiles, biophysical and sequence properties, conserved gene neighbourhoods, and translation efficiency profiles) for approximately 5,000 bacterial genomes. Combining predictions across feature sets via late fusion yielded results competitive with the state of the art [147]. Extending this to metagenome phyletic profiles — derived from sequencing environmental samples rather than isolated organisms — made it possible to predict hundreds of additional gene functions not reachable from individual genome data alone [148].

A recent application used machine learning to assist the selection of peptide insertion sites for regulating protein function [110], in collaboration with the National Institute of Chemistry. Semi-supervised learning proved instrumental in this case, which has important practical implications for designing personalized gene therapies. We have also applied multi-target prediction methods to relate environmental factors to microbiota structure — in chicken and human gut (bacterial) microbiota [122, 123], as well as fungal communities (mycobiota) in beach sand [95] — a set of problems that sit at the boundary between the life and environmental sciences.

The methods for automated scientific modelling described in Section 4 have also been applied in the life sciences. One application concerns the modelling of Rab5-to-

Rab7 conversion that governs endosome maturation during endocytosis, a key process in the immune response. ProBMoT [26] was used to select among candidate model structures encoding different hypotheses about protein domain interactions, and fit their parameters to time-series data; the best-performing structure corresponds to a cut-out/toggle switch mechanism consistent with biological understanding [133]. The framework was subsequently extended to stochastic process-based models [134] and applied in the epidemiological domain, where compartmental models of the SIR and SLIR families were learned from outbreak data for the Eyam plague and the Tristan da Cunha influenza epidemic: A more complex SLIAQRS model was later fitted to the COVID-19 epidemic in Slovenia.

6.2 Environmental Sciences

Environmental science was one of the earliest and most productive application domains for the multi-target prediction methods developed in our group. Also, it was the needs of this domain that originally motivated many of the methodological developments described in Sections 2 and 4. The central problem in many environmental settings is to relate the properties of the environment to the composition of the biota present at a site — or, conversely, to infer environmental state from biological indicators.

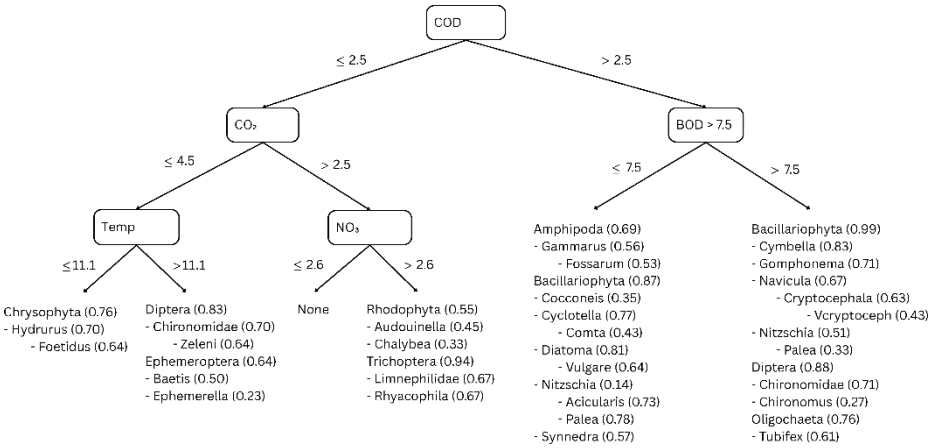

Fig. 5. A decision tree for hierarchical multi-label classification, predicting the taxonomic structure of the biota in Slovenian rivers from physical and chemical parameters, such as COD (chemical oxygen demand), placed in the internal nodes of the trees. The leaves predict the hierarchy of taxa expected to appear (with their probabilities).

The earliest such application in our work concerned Slovenian rivers [37]: Here, data from 1,060 sampling sites, described by 16 physical and chemical water properties, were used to predict the presence and abundance of nearly 500 bioindicator species [77]. Rather than building separate habitat models for each species, we developed multi-target classification trees that predict the entire community composition simultaneously. Extending this to hierarchical multi-label classification (HMLC) allowed us

to predict not only individual species, but the full taxonomic structure of the community — genera, families, and orders — from environmental variables alone, as shown in the decision tree for HMLC in Figure 5.

The same methodology was subsequently applied to soil microarthropods on Danish farms (1,944 samples, 35 collembolan species) and to vegetation at nearly 28,000 sites across Victoria, Australia (81 environmental attributes, over 3,000 species). In all cases, joint modelling of all species simultaneously yielded better average predictive performance than modelling each species separately, in addition to producing more interpretable models that describe conditions for the whole community rather than for individual species in isolation. Using the full taxonomic structure of the community in HMLC, further improved predictive performance [80].

We have also worked in the reverse direction, estimating environmental state from biological data, e.g., predicting chemical and physical parameters of river water from the presence of bioindicator organisms [36]. In a similar spirit, we have estimated ecosystem state from remotely sensed data, applying multi-target regression tree ensembles to predict canopy cover, stand height, and vertical vegetation profiles of Slovenian forests from multi-temporal, multi-spectral satellite images, using LiDAR measurements as ground truth [130]. The predictive models were applied to the entire Karst region of Slovenia, producing maps of forest height and density, useful for forest management and biomass estimation in carbon accounting, and actually used in a downstream application of fire risk assessment.

We developed fire risk models, separately for three regions of Slovenia (the Karst, the Coastal region, and continental Slovenia) using decision trees, classification rules, and tree ensembles trained on historical fire data combined with GIS inputs and meteorological forecasts [129]. For the Karst region, the data on the state of the forest (as estimated by the models described above) were also used in training the fires risk models. The learned fire risk models were deployed in the national system e-GIS UJME for natural disasters, used by the fire-fighters association, the Administration for Civil Protection and Rescue, the Environment Agency, and the Forest Service — an example of models learned from ecological data reaching deployment.

An important class of applications concerns the environmental impact of agriculture. Multi-target regression trees were used to map soil resilience in Scotland at a national level, predicting resistance and resilience characteristics of soils under both biological and physical stress [29]. Kuzmanovski et al. [70] address the task of predicting water outflows over (surface runoff) and through (discharge through sub-surface drainage systems) agricultural soils in France. Nikoloski et al. [94] use semi-supervised multi-target regression trees to predict biological water quality and the concentrations of phosphorus and nitrogen in receiving water bodies from agricultural pressure-pathway variables, such as fertilizer inputs, soil drainage, and surface runoff.

The process-based modelling methods, briefly described in Section 4 (and discussed in more detail by Langley [72] and Radinja et al. [113], both in this volume) have also found important application in the environmental sciences. A first class of applications concerns the modelling of population dynamics in aquatic ecosystems. A library of domain knowledge encoding the entities and processes typical of such ecosystems — primary producers, consumers, nutrients, and their interactions — was developed by

Atanasova et al. [8] and used to automatically build models for a wide range of aquatic systems, including the Lagoon of Venice, the Ross Sea in Antarctica [5], and lakes in Slovenia [6], Denmark, Switzerland, Japan [7], and Israel. The same library and modelling approach was applied, with modest additional effort, to each new ecosystem, demonstrating the reusability of domain knowledge across diverse settings.

A second, more recent class of applications addresses urban drainage — a problem made increasingly pressing by changing precipitation patterns due to climate change. A knowledge library for automated urban runoff modelling was developed and used with the ProBMoT tool to find optimal rainfall-runoff models for an urban catchment in Ljubljana [111]. The resulting models were then used to support the design of stormwater control measures — decentralised interventions such as green roofs and permeable surfaces — by simulating the hydrological effects of different design choices [112]. This is discussed in detail by Radinja et al. [113] (this volume).

6.3 Materials Science

Materials science presents AI with a distinctive set of challenges: datasets are typically small, experiments are expensive, and the relationship between synthesis parameters, material morphology, and functional properties is complex and only partially understood. At the same time, materials innovation is widely recognised as a bottleneck of the green transition — materials determine the performance, cost, and sustainability of products in energy storage and conversion, manufacturing and mobility, and electronics and sensing. AI-assisted materials design can significantly shorten discovery cycles by guiding experiments and simulations, and enabling self-driving laboratories.

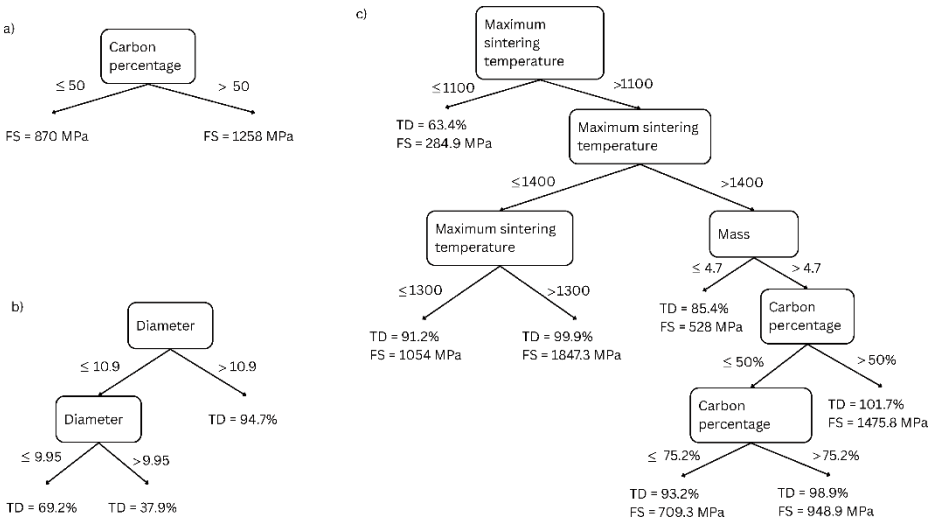

Fig. 6. Two single-target regression trees predicting a) flexural strength (FS) and b) theoretical density (TD) of tungsten-based composites, which are materials of importance for fusion power plants. The multi-target regression tree c) predicts both properties simultaneously.

Our work in this domain has centred on learning predictive models that relate synthesis parameters (and sometimes quantum-level properties) of materials to their functional properties — electrical conductivity, catalytic activity, magnetic properties, mechanical strength — and then using these models to guide materials design through optimisation. A first illustrative example [59] concerns tungsten-based composites, materials that are relevant in the context of building fusion power plants. Here the task is to predict two mechanical properties simultaneously: flexural strength (FS) and percentage of theoretical density (TD). Figure 6ab) shows two single-target regression trees, each predicting one property independently. The tree for flexural strength splits on carbon percentage alone, while the tree for theoretical density first splits on particle diameter, and for smaller particles further on diameter. Each tree captures one aspect of the material's behaviour, but neither reveals how the two properties relate to each other or which synthesis parameters govern both simultaneously.

The multi-target regression tree in Figure 6c) predicts both properties jointly, telling a different and richer story. Maximum sintering temperature emerges as the dominant variable at the root — a factor that does not appear at the top of either single-target tree. This clearly shows that sintering temperature governs the consolidation of the composite as a whole, while composition variables play secondary, conditional roles. The multi-target tree is also more directly actionable for a materials engineer seeking to simultaneously optimise multiple properties.

Our most developed application in materials science concerns foamed glass, a low-cost, non-toxic, fire-resistant, and thermally insulating material that can be produced at scale from recycled waste glass, making it relevant both economically and environmentally. We trained a neural network to predict both the density and porosity of foamed glass from synthesis parameters including glass composition, water glass content, foaming time, and furnace temperature. The trained model was then used with multi-objective optimisation to identify synthesis parameter settings that simultaneously optimise both properties, tracing out a Pareto-optimal frontier. New materials were then synthesised following the AI-generated recommendations and confirmed in laboratory tests to have the desired properties [54] — a complete cycle from data to model to new material. Semi-supervised learning also proved beneficial in this domain, as porosity measurements were missing for a subset of samples; using the partially labelled examples yielded a clear improvement in predictive performance.

More recently, we applied visual foundation models to the analysis of transmission electron microscopy images of barium hexaferrite nanoplatelets. We used the Segment Anything Model to determine particle size distributions, that in turn predict magnetic properties [88]. This is an example of how foundation models are beginning to complement classical ML approaches in materials characterisation.

7 AI Infrastructure for Science: AI Factories and SLAIF

The methods and applications described above all depend on access to appropriate computational infrastructure. Realising the full potential of AI for science requires not just algorithms and data, but also the computing power to train and deploy large models,

the data services to store and share scientific datasets, and the support services to help scientists who are not AI specialists use these tools effectively. This is the role of AI factories — a concept that has recently become a strategic priority at European level.

AI factories are large-scale infrastructure hubs that bring together computing power, data, and human capital to support the development and deployment of AI models and applications. They serve as one-stop shops for researchers, startups, industry, and public institutions seeking to harness AI. The European Commission identified AI factories as a strategic priority in its 2024 AI Innovation Package [43], and the subsequent AI Continent Action Plan [44] set out an ambitious programme to establish many AI factories across Europe, leveraging the supercomputing infrastructure of the EuroHPC Joint Undertaking. Just as the 20th century required nations to build roads and electrical power grids, the 21st century requires them to build AI infrastructure — a matter of both technological sovereignty and global competitiveness.

We have long viewed AI as infrastructure for science. Our conviction predates the current wave of enthusiasm for AI factories by many years. We established an infrastructural center on AI for science at JSI (Jožef Stefan Institute) already in 2021. Our view holds that AI infrastructure is not merely hardware: it encompasses data repositories and data management services, software platforms and tools, reusable workflows for common tasks such as model training and fine-tuning, and the human expertise needed to deploy these resources effectively. Foundation models and multimodal scientific workflows make this broader conception especially important: their effective reuse depends on model catalogues, data services, workflow orchestration, and expert support, as well as shared computational resources. Scientists wishing to apply foundation models to electron microscopy images, or to train semi-supervised tree ensembles on a partially labelled ecological dataset, need more than raw computing power — they need curated data, appropriate workflows, domain-adapted models, and professional support. Providing this full stack of resources is what distinguishes an AI factory from a conventional HPC centre and makes it genuinely useful to the scientific community.

Slovenia participates in the European AI factory initiative through the Slovenian AI Factory, SLAIF, selected by EuroHPC JU in March 2025 as one of the second cohort of European AI factories. SLAIF (www.slaif.si) consists of two interlocking consortia. The AI HPC Consortium — comprising IZUM as the EuroHPC hosting entity, JSI, and ARNES — provides the underlying AI-optimised supercomputing infrastructure, building on the Vega HPC system. The AI Factory Consortium —led technically by JSI and coordinated administratively by IZUM, and involving the Universities of Ljubljana, Maribor, Nova Gorica, and Primorska, the Faculty of Information Studies Nova Mesto, ARNES, the Chamber of Commerce and Industry of Slovenia, and Technology Park Ljubljana — delivers the services and applications built on top of that infrastructure. This broad consortium brings together Slovenia's leading research and educational institutions, HPC infrastructure operators, and industry support organisations, creating a national AI ecosystem that serves users across sectors and levels of AI readiness.

SLAIF provides access to computing and data infrastructure, general workflows for training and fine-tuning foundation models — including large language, speech, vision-language, and multimodal models — and domain-specific vertical services tailored to four strategic sectors. The SLAIF are Green Transition (incl. precision agriculture and

environmental monitoring using EO data), Health and Biotechnology (incl. personalised medicine and drug discovery), Digital Society (focusing on language technologies for Slovene and generative AI tools for the creative/ media sectors, public administration/ education) and AI for Science. The latter supports applications in life & environmental sciences, materials science and interdisciplinary research, emphasising automated scientific model discovery and facilitating reuse/sharing of scientific knowledge.

For the scientific community, the AI for Science vertical offers more than compute access. It provides domain-specific workflows aligned with the methods described in this chapter — from automated modelling pipelines to explainable ML frameworks — alongside curated scientific datasets, connections to the European Open Science Cloud (EOSC) and other data spaces, and professional support for scientists at all levels of AI expertise. SLAIF also has a strong education and training component. SLAIF thus represents both a national investment in AI infrastructure and a concrete mechanism for making the AI methods described in this chapter accessible to the broader scientific community in Slovenia and beyond.

8 Conclusions

This chapter presented a range of AI methods developed with the specific demands of scientific work in mind, organised around four broad themes: explainable machine learning, foundation models, automated scientific modelling, and semantic technologies for open science. Representative applications in life sciences, environmental sciences, and materials science have illustrated these methods in practice, and the Slovenian AI Factory has been presented as an example of how AI infrastructure can make these capabilities accessible to the broader scientific community.

The methodological landscape covered here is unified by a set of principles that were articulated in the early 2000s by the computational scientific discovery community and remain as relevant today as when they were first formulated: AI systems for science should produce outputs that are understandable to domain scientists, exploit existing background knowledge, learn from small datasets, produce models that explain data rather than merely predict it, and support productive interaction with the scientists who use them. The methods presented in this chapter — from multi-target prediction trees through process-based modelling to semantic technologies — were all developed with these principles in mind. Foundation models represent a complementary paradigm rather than a replacement: they address the data-scarce regime through transfer learning, but the outputs they produce still need to be interpretable and scientifically grounded to be genuinely useful.

Looking forward, the field of AI for science faces both extraordinary opportunities and real risks. On the opportunity side, the convergence of methodological maturity, growing availability of scientific data, and the emergence of AI infrastructure — AI factories and the broader ecosystem around them — creates conditions that did not exist a decade ago. Scientists across disciplines have access to computational resources and AI tools that were previously only accessible to large well-funded groups. Foundation models are beginning to complement classical ML approaches in materials design, drug

discovery & Earth observation. Automated scientific modelling is moving from proof-of-concept applications to practical deployment. The field is genuinely accelerating.

At the same time, the speed of development brings risks that deserve explicit attention. One is the risk of divergence: as the field expands rapidly and attracts researchers from many different communities, there is a real danger of fragmentation — of methods being developed without awareness of closely related prior work, of wheels being reinvented, of evaluation standards diverging to the point where results become incomparable. The other is the risk of forgetting where we come from: the hard-won lessons of computational scientific discovery — on the importance of explainability, domain knowledge, and small-data learning — can be lost when the field is flooded with researchers with primary experience in applications where these constraints do not apply.

Addressing both risks requires deliberate community-building. It is important to maintain and strengthen the communities that work at the intersection of AI and the individual scientific disciplines. It is these communities where AI method developers and domain scientists interact regularly, where evaluation is grounded in real scientific questions, and where the feedback loop between method development and scientific use is kept short and active.

The growing recognition of this need is visible in the emergence of dedicated conference venues. The International Conference on AI for Science, first held in Ljubljana in September 2025 [2], brought together the 28th Discovery Science Conference [35] — focused on AI methods for science — with a series of thematic tracks on AI and individual scientific disciplines: materials science, physics, life sciences, environmental science, digital humanities, and space. The structure of this event was deliberately designed to foster interaction between the two communities: AI method developers on one side, and domain scientists who have learned to use AI on the other. These two communities often develop in parallel without sufficient cross-pollination — the former may be unaware of the most pressing scientific needs, the latter may be unaware of the most powerful available methods. The second edition of this conference takes place in Mainz in October 2026 [1], where the Discovery Science conference takes place together with four disciplinary tracks: physics, chemistry and materials, life sciences, and humanities and social sciences. That this series already has a second edition, in a different country and with a broader disciplinary scope, suggests the community is responding to a genuine need.

The scientific community is not the only one responding. At the policy level, the European Commission has identified AI for science as a strategic priority in its own right, complementing the AI Factories initiative described in Section 7. The AI Continent Action Plan [44] includes a dedicated AI in Science Strategy [42], and the Commission has launched RAISE — the Resource for AI Science in Europe — as a concrete funding and coordination mechanism bringing together compute, data, and talent in support of scientific research. The inaugural AI in Science Summit (AIS25), held in Copenhagen in November 2025 under the Danish EU Presidency [3], served as the launch event for RAISE and brought together scientists, industry leaders, investors, and policymakers to explore how Europe can drive the transformation of scientific discovery through AI in a responsible and distinctive way. Thematic workshops addressed life sciences, materials science, planet and climate, and the relationship between science

and AI development itself. A second edition, AIS26, will take place under the Irish EU Presidency in 2026 [57]. That an event of this kind — high-level, cross-disciplinary, and explicitly linking scientific and policy communities — now has a recurring annual format is a clear signal that AI for science has moved from a niche research agenda to a mainstream European priority.

Amid these institutional developments, the feedback loop between AI method development and use in science remains, I believe, the most important structural feature of productive AI-for-science research: it is what ensures that methods are developed in response to genuine scientific need, that their outputs are interpretable by the scientists who must act on them, and that their limitations are discovered and addressed through real-world use rather than benchmark performance alone. Keeping this loop alive — across the full range of scientific disciplines, and across the growing divide between those who build AI infrastructure and those who use it for science — is a central challenge for the community in the years ahead.

Additional Materials. An outline of this chapter emerged after the author delivered a keynote talk [33] titled "Artificial Intelligence for Science" at ECML/PKDD 2025, The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, held in Porto, Portugal from 15th to 19th September 2025. The abstract and slides of the talk are available from the conference website [32]. A full-day course on "Artificial Intelligence for Science" was also taught by the author on 20th May 2026 as an activity of the Slovenian AI Factory (SLAIF): The slides for and recordings of the lectures are available through http://ai4science.ijs.si/2026course/ [31].

Acknowledgments. The work presented herein spans more than three decades and has been performed in collaboration with a wide range of colleagues. I would like to thank first and foremost my closest collaborators on the topics covered in this chapter: Ljupčo Todorovski and Nataša Atanasova for automated scientific modelling; Dragi Kocev, Hendrik Blockeel, Michelangelo Ceci and Jurica Levatić for explainable machine learning; Panče Panov and Larisa Soldatova for semantic technologies; and Matej Martinc and Tome Eftimov for foundation models. I would then like to thank my colleagues that have been an inspiration in developing this research agenda, including Ivan Bratko, Nada Lavrač, and Luc De Raedt; Stephen Muggleton and Ross King; and Pat Langley. I would next like to thank my numerous PhD students and postdocs that have brought the broad research agenda presented here to life, through developing a myriad of machine learning methods and using them on many scientific problems across a range of domains. Finally, I would like to thank my countless collaborators and colleagues from different scientific disciplines that have been essential in building the bridges across the AI community and their fields of research, a key prerequisite for the successful use of AI in science.

Funding acknowledgements. Given the broad scope and wide temporal range of the work presented here, it has been supported, in the past and present, by a plethora of projects. The author is currently supported, among other, through the grant GC-0001 (AI for Science) and the program P2-0103 (Knowledge Technologies) by ARIS, the Slovenian Research and Innovation Agency and the HE Grant SLAIF 101254461 (funded by EuroHPC JU and Slovenian Ministry of Education, Science and Youth).

Disclosure of Interests. The author has no competing interests to declare that are relevant to the content of this article.

References

1. AI for Science: AI4Sci - Second International Conference on AI for Science (AI4Sci 2026), https://ai4sci.eu, last accessed 2026/07/31.
2. AI for Science: International Conference on AI for Science 2025, https://ai4science.si/, last accessed 2026/07/31.
3. AI in Science Summit: AI in Science Summit 2025 (AIS25), https://ais25.eu/, last accessed 2026/07/31.
4. Arp, R. et al.: Building ontologies with basic formal ontology. The MIT Press, Cambridge, Massachusetts (2015).
5. Asgharbeygi, N. et al.: Inductive revision of quantitative process models. Ecological Modelling. 194, 1–3, 70–79 (2006). https://doi.org/10.1016/j.ecolmodel.2005.10.008.
6. Atanasova, N. et al.: Automated modelling of a food web in lake Bled using measured data and a library of domain knowledge. Ecological Modelling. 194, 1–3, 37–48 (2006). https://doi.org/10.1016/j.ecolmodel.2005.10.029.
7. Atanasova, N. et al.: Computational Assemblage of Ordinary Differential Equations for Chlorophyll-a Using a Lake Process Equation Library and Measured Data of Lake Kasumigaura. In: Recknagel, F. (ed.) Ecological Informatics. pp. 409–427 Springer Berlin (2006). https://doi.org/10.1007/3-540-28426-5_20.
8. Atanasova, N. et al.: Constructing a library of domain knowledge for automated modelling of aquatic ecosystems. Ecological Modelling. 194, 1–3, 14–36 (2006). https://doi.org/10.1016/j.ecolmodel.2005.10.002.
9. Barredo Arrieta, A. et al.: Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion. 58, 82–115 (2020). https://doi.org/10.1016/j.inffus.2019.12.012.
10. Blockeel, H. et al.: Top-Down Induction of Clustering Trees. In: Proceedings of the 15th International Conference on Machine Learning (ICML 1998). pp. 55–63 Morgan Kaufmann, San Francisco, CA (1998).
11. Blockeel, H., De Raedt, L.: Top-down induction of first-order logical decision trees. Artificial Intelligence. 101, 1–2, 285–297 (1998). https://doi.org/10.1016/S0004-3702(98)00034-4.
12. Bogatinovski, J. et al.: Explaining the performance of multilabel classification methods with data set properties. International Journal of Intelligent Systems. 37, 9, 6080–6122 (2022). https://doi.org/10.1002/int.22835.
13. Boiko, D.A. et al.: Autonomous chemical research with large language models. Nature. 624, 7992, 570–578 (2023). https://doi.org/10.1038/s41586-023-06792-0.
14. Bommasani, R. et al.: On the Opportunities and Risks of Foundation Models, (2021). https://doi.org/10.48550/arxiv.2108.07258.

15. Bran, A.M. et al.: Augmenting large language models with chemistry tools. Nature Machine Intelligence. 6, 5, 525–535 (2024). https://doi.org/10.1038/s42256-024-00832-8.
16. Breiman, L. et al.: Classification and regression trees. Wadsworth International Group, Belmont, CA (1984).
17. Breiman, L.: Random Forests. Machine Learning. 45, 1, 5–32 (2001). https://doi.org/10.1023/A:1010933404324.
18. Brence, J. et al.: Probabilistic grammars for equation discovery. Knowledge-Based Systems. 224, 107077 (2021). https://doi.org/10.1016/j.knosys.2021.107077.
19. Breskvar, M. et al.: Relating Biological and Clinical Features of Alzheimer's Patients with Predictive Clustering Trees. In: Proceedings of the 18th International Multiconference Information Society (IS 2015), Volume E. pp. 9–12 Jožef Stefan Institute, Ljubljana, Slovenia (2015).
20. Bridewell, W. et al.: Inductive process modeling. Machine Learning. 71, 1, 1–32 (2008). https://doi.org/10.1007/s10994-007-5042-6.
21. Brown, T.B. et al.: Language Models are Few-Shot Learners. In: Advances in Neural Information Processing Systems 33 (NeurIPS 2020). pp. 1877–1901 (2020).
22. Brugger, J. et al.: Neural-Guided Equation Discovery. In: Džeroski, S. et al. (eds.) Artificial Intelligence for Science: Computational Approaches to Scientific Discovery. pp. 178–215 Springer, Berlin (2026).
23. Brunton, S.L. et al.: Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proceedings of the National Academy of Sciences. 113, 15, 3932–3937 (2016). https://doi.org/10.1073/pnas.1517384113.
24. Buchanan, B.G. et al.: Heuristic DENDRAL: A Program for Generating Explanatory Hypotheses in Organic Chemistry. In: Meltzer, B. and Michie, D. (eds.) Machine Intelligence 4. pp. 209–254 Edinburgh University Press, Edinburgh (1969).
25. Burger, B. et al.: A mobile robotic chemist. Nature. 583, 7815, 237–241 (2020). https://doi.org/10.1038/s41586-020-2442-2.
26. Čerepnalkoski, D. et al.: The influence of parameter fitting methods on model structure selection in automated modeling of aquatic ecosystems. Ecological Modelling. 245, 136–165 (2012). https://doi.org/10.1016/j.ecolmodel.2012.06.001.
27. Chaushevska, M. et al.: Learning the Probabilities in Probabilistic Context-Free Grammars for Arithmetical Expressions from Equation Corpora. In: Proceedings of the 25th International Multiconference Information Society – IS 2022, Volume A: Slovenian Conference on Artificial Intelligence. pp. 11–14 Jožef Stefan Institute, Ljubljana (2022).
28. Cui, H. et al.: Towards multimodal foundation models in molecular cell biology. Nature. 640, 8059, 623–633 (2025). https://doi.org/10.1038/s41586-025-08710-y.

29. Debeljak, M. et al.: Potential of multi-objective models for risk-based mapping of the resilience characteristics of soils: demonstration at a national level. Soil Use and Management. 25, 1, 66–77 (2009). https://doi.org/10.1111/j.1475-2743.2009.00196.x.
30. Devlin, J. et al.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Association for Computational Linguistics. pp. 4171–4186 , Minneapolis, Minnesota (2019). https://doi.org/10.18653/v1/N19-1423.
31. Džeroski, S.: Artificial Intelligence for Science - Course Materials, http://ai4science.ijs.si/2026course/.
32. Džeroski, S.: Artificial Intelligence for Science - Keynote Talk Slides, https://ecmlpkdd.org/2025/keynotes/.
33. Džeroski, S.: Artificial Intelligence for Science: Keynote talk. In: Ribeiro, R.P. et al. (eds.) Machine learning and knowledge discovery in databases: European Conference, ECML PKDD 2025, Porto, Portugal, September 15-19, 2025: Proceedings, Research track, Part I. p. lxxii Springer, Cham (2026).
34. Džeroski, S. et al.: Computational Discovery of Scientific Knowledge. In: Džeroski, S. and Todorovski, L. (eds.) Computational Discovery of Scientific Knowledge. pp. 1–14 Springer, Berlin (2007). https://doi.org/10.1007/978-3-540-73920-3_1.
35. Džeroski, S. et al. eds: Discovery Science: 28th International Conference, DS 2025, Ljubljana, Slovenia, September 23–25, 2025, Proceedings. Springer, Cham (2025). https://doi.org/10.1007/978-3-032-05461-6.
36. Džeroski, S. et al.: Predicting Chemical Parameters of River Water Quality from Bioindicator Data. Applied Intelligence. 13, 1, 7–17 (2000). https://doi.org/10.1023/A:1008323212047.
37. Džeroski, S., Grbović, J.: Knowledge Discovery in a Water Quality Database. In: Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95). pp. 81–86 AAAI Press, Menlo Park, CA (1995).
38. Džeroski, S., Petrovski, I.: Discovering dynamics with genetic programming. In: Bergadano, F. and Raedt, L. (eds.) Machine Learning: ECML-94. pp. 347–350 Springer Berlin (1994). https://doi.org/10.1007/3-540-57868-4_70.
39. Džeroski, S., Todorovski, L.: Discovering Dynamics. In: Proceedings of the 10th International Conference on Machine Learning (ICML 1993). pp. 97–103 Morgan Kaufmann, San Mateo, CA (1993).
40. Džeroski, S., Todorovski, L.: Encoding and Using Domain Knowledge on Population Dynamics for Equation Discovery. In: Magnani, L. et al. (eds.) Logical and Computational Aspects of Model-Based Reasoning. pp. 227–247 Kluwer, Dordrecht (2002). https://doi.org/10.1007/978-94-010-0550-0_11.
41. Edgar, R.: Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. Nucleic Acids Research. 30, 1, 207–210 (2002). https://doi.org/10.1093/nar/30.1.207.
42. European Commission: A European Strategy for Artificial Intelligence in Science: Paving the Way for the Resource for AI Science in Europe (RAISE), COM

(2025) 724 final, https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52025DC0724.
43. European Commission: Communication on Boosting Startups and Innovation in Trustworthy Artificial Intelligence, COM (2024) 28 final, https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52024DC0028.
44. European Commission: The AI Continent Action Plan, COM (2025) 165 final, https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52025DC0165.
45. Gama, J.: Knowledge discovery from data streams. Chapman & Hall/CRC, Boca Raton, FL (2010).
46. Gjorgjevikj, A. et al.: Large language models in food and nutrition science: Opportunities, challenges, and the case of FoodyLLM. Current Research in Food Science. 12, 101351 (2026). https://doi.org/10.1016/j.crfs.2026.101351.
47. Gower, A.H. et al.: The Use of AI-Robotic Systems for Scientific Discovery. In: Džeroski, S. et al. (eds.) Artificial Intelligence for Science: Computational Approaches to Scientific Discovery. pp. 99–121 Springer, Berlin (2026).
48. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition. 5, 2, 199–220 (1993). https://doi.org/10.1006/knac.1993.1008.
49. Guidotti, R. et al.: A Survey of Methods for Explaining Black Box Models. ACM Computing Surveys. 51, 5, 1–42 (2018). https://doi.org/10.1145/3236009.
50. Gunning, D. et al.: XAI—Explainable artificial intelligence. Science Robotics. 4, 37, eaay7120 (2019). https://doi.org/10.1126/scirobotics.aay7120.
51. Hayes, C.F. et al.: Deep Symbolic Optimization: Reinforcement Learning for Symbolic Mathematics. In: Džeroski, S. et al. (eds.) Artificial Intelligence for Science: Computational Approaches to Scientific Discovery. pp. 147–177 Springer, Berlin (2026).
52. He, K. et al.: Masked Autoencoders Are Scalable Vision Learners. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16000–16009 IEEE, New Orleans, LA, USA (2022). https://doi.org/10.1109/CVPR52688.2022.01553.
53. Hinton, G.E., Salakhutdinov, R.R.: Reducing the Dimensionality of Data with Neural Networks. Science. 313, 5786, 504–507 (2006). https://doi.org/10.1126/science.1127647.
54. Hribar, U. et al.: Optimizing foamed glass production with machine learning. Materials & Design. 257, 114459 (2025). https://doi.org/10.1016/j.matdes.2025.114459.
55. Hu, E.J. et al.: LoRA: Low-Rank Adaptation of Large Language Models. In: Proceedings of the Tenth International Conference on Learning Representations (ICLR). (2022).
56. Hutter, F. et al. eds: Automated Machine Learning: Methods, Systems, Challenges. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05318-5.
57. Irish Presidency of the Council of the European Union: AI in Science Summit 2026 (AIS26), https://irish-presidency.consilium.europa.eu/en/events/ai-in-science-summit-2026-ais26/, last accessed 2026/07/31.

58. Jakubik, J. et al.: TerraMind: Large-Scale Generative Multimodality for Earth Observation. In: 2025 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 7383–7394 IEEE, Honolulu, HI, USA (2025). https://doi.org/10.1109/ICCV51701.2025.00693.
59. Jenuš, P. et al.: Relating the Composition and Mechanical Properties of Tungsten-Based Composites by Using Machine Learning. In: ML4M 2022 - Young Researchers' Workshop on Machine Learning for Materials. p. 33 , SISSA Miramare Campus, Trieste, Italy (2022).
60. King, R.D. et al.: Functional genomic hypothesis generation and experimentation by a robot scientist. Nature. 427, 6971, 247–252 (2004). https://doi.org/10.1038/nature02236.
61. King, R.D. et al.: The Automation of Science. Science. 324, 5923, 85–89 (2009). https://doi.org/10.1126/science.1165620.
62. Kirillov, A. et al.: Segment Anything. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 4015–4026 IEEE, Paris, France (2023). https://doi.org/10.1109/ICCV51070.2023.00371.
63. Kocev, D. et al.: Tree ensembles for predicting structured outputs. Pattern Recognition. 46, 3, 817–833 (2013). https://doi.org/10.1016/j.patcog.2012.09.023.
64. Koloski, B. et al.: Digitalisation of inorganic chemistry with LLMs. Inorganic Chemistry Frontiers. 13, 8, 3213–3219 (2026). https://doi.org/10.1039/D6QI00240D.
65. Kompare, B. et al.: Modelling the Growth of Algae in the Lagoon of Venice with the Artificial Intelligence Tool GoldHorn. In: Proceedings of the Fourth International Conference on Water Pollution. pp. 799–808 Computational Mechanics Publications, Southampton (1997).
66. Kononenko, I.: Estimating attributes: Analysis and extensions of RELIEF. In: Bergadano, F. and Raedt, L. (eds.) Machine Learning: ECML-94. pp. 171–182 Springer Berlin (1994). https://doi.org/10.1007/3-540-57868-4_57.
67. Korbee, C.J. et al.: Combined chemical genetics and data-driven bioinformatics approach identifies receptor tyrosine kinase inhibitors as host-directed antimicrobials. Nature Communications. 9, 1, 358 (2018). https://doi.org/10.1038/s41467-017-02777-6.
68. Kostovska, A. et al.: OPTION: OPTImization Algorithm Benchmarking ONtology. IEEE Transactions on Evolutionary Computation. 27, 6, 1618–1632 (2023). https://doi.org/10.1109/TEVC.2022.3232844.
69. Koza, J.R.: Genetic programming. 1: On the programming of computers by means of natural selection. MIT Press, Cambridge, MA (1992).
70. Kuzmanovski, V. et al.: Modeling water outflow from tile-drained agricultural fields. Science of The Total Environment. 505, 390–401 (2015). https://doi.org/10.1016/j.scitotenv.2014.10.009.
71. Langley, P.: BACON: A Production System that Discovers Empirical Laws. In: Proceedings of the Fifth International Joint Conference on Artificial Intelligence (IJCAI 1977). p. 344 William Kaufmann, Cambridge, MA (1977).

72. Langley, P.: Computational Discovery of Quantitative Process Models. In: Džeroski, S. et al. (eds.) Artificial Intelligence for Science: Computational Approaches to Scientific Discovery. pp. 261–283 Springer, Berlin (2026).
73. Langley, P. et al.: Inducing Process Models from Continuous Data. In: Proceedings of the Nineteenth International Conference on Machine Learning (ICML 2002). pp. 347–354 Morgan Kaufmann, San Francisco (2002).
74. Langley, P.: Lessons for the Computational Discovery of Scientific Knowledge. In: Proceedings of the First International Workshop on Data Mining Lessons Learned (held in conjunction with the 19th International Conference on Machine Learning, ICML 2002). pp. 9–12 , Sydney, Australia (2002).
75. Langley, P.: Scientific discovery, causal explanation, and process model induction. Mind & Society. 18, 1, 43–56 (2019). https://doi.org/10.1007/s11299-019-00216-1.
76. Langley, P. et al.: Scientific Discovery: Computational Explorations of the Creative Processes. MIT Press, Cambridge (1987).
77. Levatić, J. et al.: Community structure models are improved by exploiting taxonomic rank with predictive clustering trees. Ecological Modelling. 306, 294–304 (2015). https://doi.org/10.1016/j.ecolmodel.2014.10.023.
78. Levatić, J. et al.: Semi-supervised classification trees. Journal of Intelligent Information Systems. 49, 3, 461–486 (2017). https://doi.org/10.1007/s10844-017-0457-4.
79. Levatić, J. et al.: Semi-Supervised Predictive Clustering Trees for (Hierarchical) Multi-Label Classification. International Journal of Intelligent Systems. 2024, 1–21 (2024). https://doi.org/10.1155/2024/5610291.
80. Levatić, J. et al.: The importance of the label hierarchy in hierarchical multi-label classification. Journal of Intelligent Information Systems. 45, 2, 247–271 (2015). https://doi.org/10.1007/s10844-014-0347-y.
81. Lewis, P. et al.: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In: Advances in Neural Information Processing Systems 33 (NeurIPS 2020). pp. 9459–9474 (2020).
82. Li, C. et al.: BioModels Database: An enhanced, curated and annotated resource for published quantitative kinetic models. BMC Systems Biology. 4, 1, 92 (2010). https://doi.org/10.1186/1752-0509-4-92.
83. Li, Y. et al.: ChatSR: Multimodal Large Language Models for Scientific Formula Discovery, (2024). https://doi.org/10.48550/arxiv.2406.05410.
84. Lindsay, R.K. et al.: DENDRAL: A case study of the first expert system for scientific hypothesis formation. Artificial Intelligence. 61, 2, 209–261 (1993). https://doi.org/10.1016/0004-3702(93)90068-M.
85. Liu, X. et al.: Self-supervised Learning: Generative or Contrastive. IEEE Transactions on Knowledge and Data Engineering. 35, 1, 857–876 (2023). https://doi.org/10.1109/TKDE.2021.3090866.
86. Lu, C. et al.: Towards end-to-end automation of AI research. Nature. 651, 8107, 914–919 (2026). https://doi.org/10.1038/s41586-026-10265-5.

87. Lundberg, S.M., Lee, S.-I.: A Unified Approach to Interpreting Model Predictions. In: Advances in Neural Information Processing Systems 30 (NeurIPS 2017). pp. 4765–4774 (2017).
88. Martinc, M. et al.: Can Segment Anything Model Segment Just Some Things? Improving Zero-Shot Segmentation of Electron Microscopic Images with Clustering. In: 2025 13th European Workshop on Visual Information Processing (EUVIP). pp. 1–6 IEEE, Valletta, Malta (2025). https://doi.org/10.1109/EUVIP66349.2025.11238970.
89. Merler, M. et al.: In-Context Symbolic Regression: Leveraging Large Language Models for Function Discovery. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop). pp. 427–444 Association for Computational Linguistics, Bangkok, Thailand (2024). https://doi.org/10.18653/v1/2024.acl-srw.49.
90. Mežnar, S. et al.: Efficient generator of mathematical expressions for symbolic regression. Machine Learning. 112, 11, 4563–4596 (2023). https://doi.org/10.1007/s10994-023-06400-2.
91. Mežnar, S. et al.: Generative Models for Equation Discovery. In: Džeroski, S. et al. (eds.) Artificial Intelligence for Science: Computational Approaches to Scientific Discovery. pp. 216–239 Springer, Berlin (2026).
92. Molnar, C. et al.: General Pitfalls of Model-Agnostic Interpretation Methods for Machine Learning Models. In: Holzinger, A. et al. (eds.) xxAI - Beyond Explainable AI. pp. 39–68 Springer, Cham (2022). https://doi.org/10.1007/978-3-031-04083-2_4.
93. Nasim, M. et al.: Expediting Symbolic Regression for Science Using Scientific Approaches. In: Džeroski, S. et al. (eds.) Artificial Intelligence for Science: Computational Approaches to Scientific Discovery. pp. 125–146 Springer, Berlin (2026).
94. Nikoloski, S. et al.: Exploiting partially-labeled data in learning predictive clustering trees for multi-target regression: A case study of water quality assessment in Ireland. Ecological Informatics. 61, 101161 (2021). https://doi.org/10.1016/j.ecoinf.2020.101161.
95. Novak Babič, M. et al.: Occurrence, Diversity and Anti-Fungal Resistance of Fungi in Sand of an Urban Beach in Slovenia—Environmental Monitoring with Possible Health Risk Implications. Journal of Fungi. 8, 8, 860 (2022). https://doi.org/10.3390/jof8080860.
96. Osojnik, A. et al.: Incremental predictive clustering trees for online semi-supervised multi-target regression. Machine Learning. 109, 11, 2121–2139 (2020). https://doi.org/10.1007/s10994-020-05918-z.
97. Osojnik, A. et al.: iSOUP-SymRF: Symbolic feature ranking with random forests in online multi-target regression and multi-label classification. Machine Learning. 114, 2, 34 (2025). https://doi.org/10.1007/s10994-024-06718-5.
98. Osojnik, A. et al.: Multi-label classification via multi-target regression on data streams. Machine Learning. 106, 6, 745–770 (2017). https://doi.org/10.1007/s10994-016-5613-5.

99. Osojnik, A. et al.: Tree-based methods for online multi-target regression. Journal of Intelligent Information Systems. 50, 2, 315–339 (2018). https://doi.org/10.1007/s10844-017-0462-7.
100. Ouyang, L. et al.: Training Language Models to Follow Instructions with Human Feedback. In: Advances in Neural Information Processing Systems 35 (NeurIPS 2022). pp. 27730–27744 (2022).
101. Pan, S.J., Yang, Q.: A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering. 22, 10, 1345–1359 (2010). https://doi.org/10.1109/TKDE.2009.191.
102. Panov, P. et al.: Generic ontology of datatypes. Information Sciences. 329, 900–920 (2016). https://doi.org/10.1016/j.ins.2015.08.006.
103. Panov, P. et al.: Ontology of core data mining entities. Data Mining and Knowledge Discovery. 28, 5–6, 1222–1265 (2014). https://doi.org/10.1007/s10618-014-0363-0.
104. Petković, M. et al.: CLUSplus: A decision tree-based framework for predicting structured outputs. SoftwareX. 24, 101526 (2023). https://doi.org/10.1016/j.softx.2023.101526.
105. Petković, M. et al.: Ensemble- and distance-based feature ranking for unsupervised learning. International Journal of Intelligent Systems. 36, 7, 3068–3086 (2021). https://doi.org/10.1002/int.22390.
106. Petković, M. et al.: Feature ranking for multi-target regression. Machine Learning. 109, 6, 1179–1204 (2020). https://doi.org/10.1007/s10994-019-05829-8.
107. Petković, M. et al.: Feature ranking for semi-supervised learning. Machine Learning. 112, 11, 4379–4408 (2023). https://doi.org/10.1007/s10994-022-06181-0.
108. Petković, M. et al.: Multi-label feature ranking with ensemble methods. Machine Learning. 109, 11, 2141–2159 (2020). https://doi.org/10.1007/s10994-020-05908-1.
109. Petković, M. et al.: Relational tree ensembles and feature rankings. Knowledge-Based Systems. 251, 109254 (2022). https://doi.org/10.1016/j.knosys.2022.109254.
110. Plaper, T. et al.: Designed allosteric protein logic. Cell Discovery. 10, 1, 8 (2024). https://doi.org/10.1038/s41421-023-00635-y.
111. Radinja, M. et al.: Automated modelling of urban runoff based on domain knowledge and equation discovery. Journal of Hydrology. 603, 127077 (2021). https://doi.org/10.1016/j.jhydrol.2021.127077.
112. Radinja, M. et al.: Design and Simulation of Stormwater Control Measures Using Automated Modeling. Water. 13, 16, 2268 (2021). https://doi.org/10.3390/w13162268.
113. Radinja, M. et al.: Process-Based Modelling of Natural and Urban Catchments. In: Džeroski, S. et al. (eds.) Artificial Intelligence for Science: Computational Approaches to Scientific Discovery. pp. 364–393 Springer, Berlin (2026).
114. Ribeiro, M.T. et al.: "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 ACM, San Francisco California USA (2016). https://doi.org/10.1145/2939672.2939778.
115. Ring, N.A.R. et al.: Wet-dry-wet drug screen leads to the synthesis of TS1, a novel compound reversing lung fibrosis through inhibition of myofibroblast differentiation. Cell Death & Disease. 13, 1, 2 (2022). https://doi.org/10.1038/s41419-021-04439-4.
116. Roy, B. et al.: Survival analysis with semi-supervised predictive clustering trees. Computers in Biology and Medicine. 141, 105001 (2022). https://doi.org/10.1016/j.compbiomed.2021.105001.
117. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence. 1, 5, 206–215 (2019). https://doi.org/10.1038/s42256-019-0048-x.
118. Samek, W. et al. eds: Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-28954-6.
119. Schmidt, M., Lipson, H.: Distilling Free-Form Natural Laws from Experimental Data. Science. 324, 5923, 81–85 (2009). https://doi.org/10.1126/science.1165893.
120. Selvaraju, R.R. et al.: Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 618–626 IEEE, Venice (2017). https://doi.org/10.1109/ICCV.2017.74.
121. Shojaee, P. et al.: LLM-SR: Scientific Equation Discovery via Programming with Large Language Models. In: Proceedings of the Thirteenth International Conference on Learning Representations (ICLR). (2025).
122. Skraban, J. et al.: Changes of poultry faecal microbiota associated with Clostridium difficile colonisation. Veterinary Microbiology. 165, 3–4, 416–424 (2013). https://doi.org/10.1016/j.vetmic.2013.04.014.
123. Skraban, J. et al.: Gut Microbiota Patterns Associated with Colonization of Different Clostridium difficile Ribotypes. PLoS ONE. 8, 2, e58005 (2013). https://doi.org/10.1371/journal.pone.0058005.
124. Škrlj, B. et al.: ReliefE: feature ranking in high-dimensional spaces via manifold embeddings. Machine Learning. 111, 1, 273–317 (2022). https://doi.org/10.1007/s10994-021-05998-5.
125. Škunca, N. et al.: Phyletic Profiling with Cliques of Orthologs Is Enhanced by Signatures of Paralogy Relationships. PLoS Computational Biology. 9, 1, e1002852 (2013). https://doi.org/10.1371/journal.pcbi.1002852.
126. Smith, B. et al.: The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. Nat Biotechnol. 25, 11, 1251–1255 (2007). https://doi.org/10.1038/nbt1346.
127. Soldatova, L.: Knowledge Engineering for Automated Scientific Discovery. In: Džeroski, S. et al. (eds.) Artificial Intelligence for Science: Computational Approaches to Scientific Discovery. pp. 3–20 Springer, Berlin (2026).

128. Soldatova, L.N., King, R.D.: An ontology of scientific experiments. Journal of The Royal Society Interface. 3, 11, 795–803 (2006). https://doi.org/10.1098/rsif.2006.0134.
129. Stojanova, D. et al.: Estimating the risk of fire outbreaks in the natural environment. Data Mining and Knowledge Discovery. 24, 2, 411–442 (2012). https://doi.org/10.1007/s10618-011-0213-2.
130. Stojanova, D. et al.: Estimating vegetation height and canopy cover from remotely sensed data with machine learning. Ecological Informatics. 5, 4, 256–266 (2010). https://doi.org/10.1016/j.ecoinf.2010.03.004.
131. Struyf, J., Džeroski, S.: Constraint Based Induction of Multi-objective Regression Trees. In: Bonchi, F. and Boulicaut, J.-F. (eds.) Knowledge Discovery in Inductive Databases. pp. 222–233 Springer Berlin Heidelberg, Berlin, Heidelberg (2006). https://doi.org/10.1007/11733492_13.
132. Swanson, K. et al.: The Virtual Lab of AI agents designs new SARS-CoV-2 nanobodies. Nature. 646, 8085, 716–723 (2025). https://doi.org/10.1038/s41586-025-09442-9.
133. Tanevski, J. et al.: Domain-specific model selection for structural identification of the Rab5-Rab7 dynamics in endocytosis. BMC Systems Biology. 9, 1, 31 (2015). https://doi.org/10.1186/s12918-015-0175-x.
134. Tanevski, J. et al.: Learning stochastic process-based models of dynamical systems from knowledge and data. BMC Systems Biology. 10, 1, 30 (2016). https://doi.org/10.1186/s12918-016-0273-4.
135. Tanevski, J. et al.: Process-based design of dynamical biological systems. Scientific Reports. 6, 1, 34107 (2016). https://doi.org/10.1038/srep34107.
136. Tang, Y. et al.: A multimodal large language model for materials science. Nature Machine Intelligence. 8, 4, 588–601 (2026). https://doi.org/10.1038/s42256-026-01214-y.
137. Taskin, B. et al.: Knowledge integration for physics-informed symbolic regression using pre-trained large language models. Scientific Reports. 16, 1, 1614 (2026). https://doi.org/10.1038/s41598-026-35327-6.
138. Todorovski, L. et al.: Modelling and prediction of phytoplankton growth with equation discovery. Ecological Modelling. 113, 1–3, 71–81 (1998). https://doi.org/10.1016/S0304-3800(98)00135-5.
139. Todorovski, L., Džeroski, S.: Declarative Bias in Equation Discovery. In: Proceedings of the Fourteenth International Conference on Machine Learning (ICML 1997). pp. 376–384 Morgan Kaufmann (1997).
140. Todorovski, L., Džeroski, S.: Integrating knowledge-driven and data-driven approaches to modeling. Ecological Modelling. 194, 1–3, 3–13 (2006). https://doi.org/10.1016/j.ecolmodel.2005.10.001.
141. Van Assche, A. et al.: First order random forests: Learning relational classifiers with complex aggregates. Machine Learning. 64, 1–3, 149–182 (2006). https://doi.org/10.1007/s10994-006-8713-9.
142. Van Engelen, J.E., Hoos, H.H.: A survey on semi-supervised learning. Machine Learning. 109, 2, 373–440 (2020). https://doi.org/10.1007/s10994-019-05855-6.

143. Vanschoren, J. et al.: Experiment databases: A new way to share, organize and learn from experiments. Machine Learning. 87, 2, 127–158 (2012). https://doi.org/10.1007/s10994-011-5277-0.
144. Vanschoren, J. et al.: OpenML: networked science in machine learning. ACM SIGKDD Explorations Newsletter. 15, 2, 49–60 (2014). https://doi.org/10.1145/2641190.2641198.
145. Vaswani, A. et al.: Attention is All You Need. In: Advances in Neural Information Processing Systems 30 (NeurIPS 2017). pp. 5998–6008 (2017).
146. Vens, C. et al.: Decision trees for hierarchical multi-label classification. Machine Learning. 73, 2, 185–214 (2008). https://doi.org/10.1007/s10994-008-5077-3.
147. Vidulin, V. et al.: Extensive complementarity between gene function prediction methods. Bioinformatics. 32, 23, 3645–3653 (2016). https://doi.org/10.1093/bioinformatics/btw532.
148. Vidulin, V. et al.: The evolutionary signal in metagenome phyletic profiles predicts many gene functions. Microbiome. 6, 1, 129 (2018). https://doi.org/10.1186/s40168-018-0506-4.
149. Vogler, M. et al.: Autonomous Battery Optimization by Deploying Distributed Experiments and Simulations. Advanced Energy Materials. 14, 46, 2403263 (2024). https://doi.org/10.1002/aenm.202403263.
150. Voosen, P.: The AI detectives. Science. 357, 6346, 22–27 (2017). https://doi.org/10.1126/science.357.6346.22.
151. Waegeman, W. et al.: Multi-target prediction: a unifying view on problems and methods. Data Mining and Knowledge Discovery. 33, 2, 293–324 (2019). https://doi.org/10.1007/s10618-018-0595-5.
152. Wilkinson, M.D. et al.: The FAIR Guiding Principles for scientific data management and stewardship. Scientific Data. 3, 1, 160018 (2016). https://doi.org/10.1038/sdata.2016.18.
153. Williams, K. et al.: Cheaper faster drug development validated by the repositioning of drugs against neglected tropical diseases. Journal of The Royal Society Interface. 12, 104, 20141289 (2015). https://doi.org/10.1098/rsif.2014.1289.
154. Xiong, Z. et al.: Neural Plasticity-Inspired Multimodal Foundation Model for Earth Observation, (2024). https://doi.org/10.48550/arxiv.2403.15356.
155. Yosinski, J. et al.: How Transferable Are Features in Deep Neural Networks? In: Advances in Neural Information Processing Systems 27 (NeurIPS 2014). pp. 3320–3328 (2014).
156. Ženko, B. et al.: Learning Predictive Clustering Rules. In: Bonchi, F. and Boulicaut, J.-F. (eds.) Knowledge Discovery in Inductive Databases. pp. 234–250 Springer Berlin (2006). https://doi.org/10.1007/11733492_14.
157. Zhang, Y. et al.: Exploring the role of large language models in the scientific method: from hypothesis to discovery. npj Artificial Intelligence. 1, 1, 14 (2025). https://doi.org/10.1038/s44387-025-00019-5.
158. Zhu, X.X. et al.: On the foundations of Earth foundation models. Communications Earth & Environment. 7, 1, 103 (2026). https://doi.org/10.1038/s43247-025-03127-x.