BMC Biology

REVIEW

Open Access

Machine learning in biological research: key algorithms, applications, and future directions

Md Nafis Ul Alam[1,2,3], Kiran Basava[1], Ani Chitransh[1], H. M. Abdul Fattah[1], Hector D. Garcia-Verdugo[1], Shih-Hsuan Lo[1], Tanisha Lohchab[1], Kristen M. Martinet[1], Cristian Román-Palacios[1,4*], Jhan Carlos Salazar[1,5,6] and Danielle Van Boxel[1,7]

Abstract

Machine learning is a robust framework to analyze questions using complex data in a variety of fields. We present definitions and recent applications of four key machine learning methods and discuss their advantages and challenges in biological research. Through a set of systematically selected case studies, we highlight how machine learning models have been used in a range of applications, including phylogenomics, disease prediction, and host taxonomy prediction. We identify additional potential areas of integration of machine learning into questions with biological relevance. This intersection can be further enhanced through collaboration and innovation on parallelization, interpretability, and preprocessing.

Keywords Biology, Data science, Large datasets, Learning

Background

Machine learning (ML) is a branch of artificial intelligence (AI) now standard for conducting cutting-edge research in a plethora of fields, including disciplines within biological sciences [1], Fig. 1). Although machine learning as a field has existed for decades [2], there is still significant room for new applications, especially as (1) new datasets emerge, (2) existing datasets increase in size, and (3) computational technologies improve. Here, we focus on reviewing four different ML algorithms by providing in-depth perspectives on their use based on recent relevant research across key biological disciplines.

Machine learning focuses on building computational systems that learn from data. These systems are ultimately expected to enhance their performance without explicit programming [3]. Relative to similar disciplines (e.g., statistics), ML explicitly considers the trade-offs associated with learning, such as the balance between accuracy of predictions and complexity of models, and the generalization of models (i.e., their ability to perform well on unseen data not used during the training process). ML algorithms develop models from data to make predictions rather than following static program instructions. To this end, the process of training the model on data is crucial for uncovering patterns that are not immediately evident in the data. Ultimately, a central challenge in ML involves managing the trade-off between the precision of predictions and the ability of models to generalize [4]. These trade-offs are specifically related to addressing issues such as overfitting, where a model is

Order of authorship is alphabetical.

*Correspondence:
Cristian Román-Palacios
cromanpa@arizona.edu
[1] Data Diversity Lab, College of Information Science, University of Arizona, Tucson, AZ, USA
[2] School of Plant Sciences, Arizona Genomics Institute, University of Arizona, Tucson, AZ, USA
[3] Department of Biochemistry and Molecular Biology, Plant Biotechnology Laboratory, University of Dhaka, Dhaka, Bangladesh
[4] Center for Diverse Leadership in Science, University of California, Los Angeles, CA, USA
[5] Department of Biology, Washington University, St. Louis, MO, USA
[6] Department of Neurosurgery, Mayo Clinic, Jacksonville, FL, USA
[7] Applied Math GIDP, University of Arizona, Tucson, AZ, USA

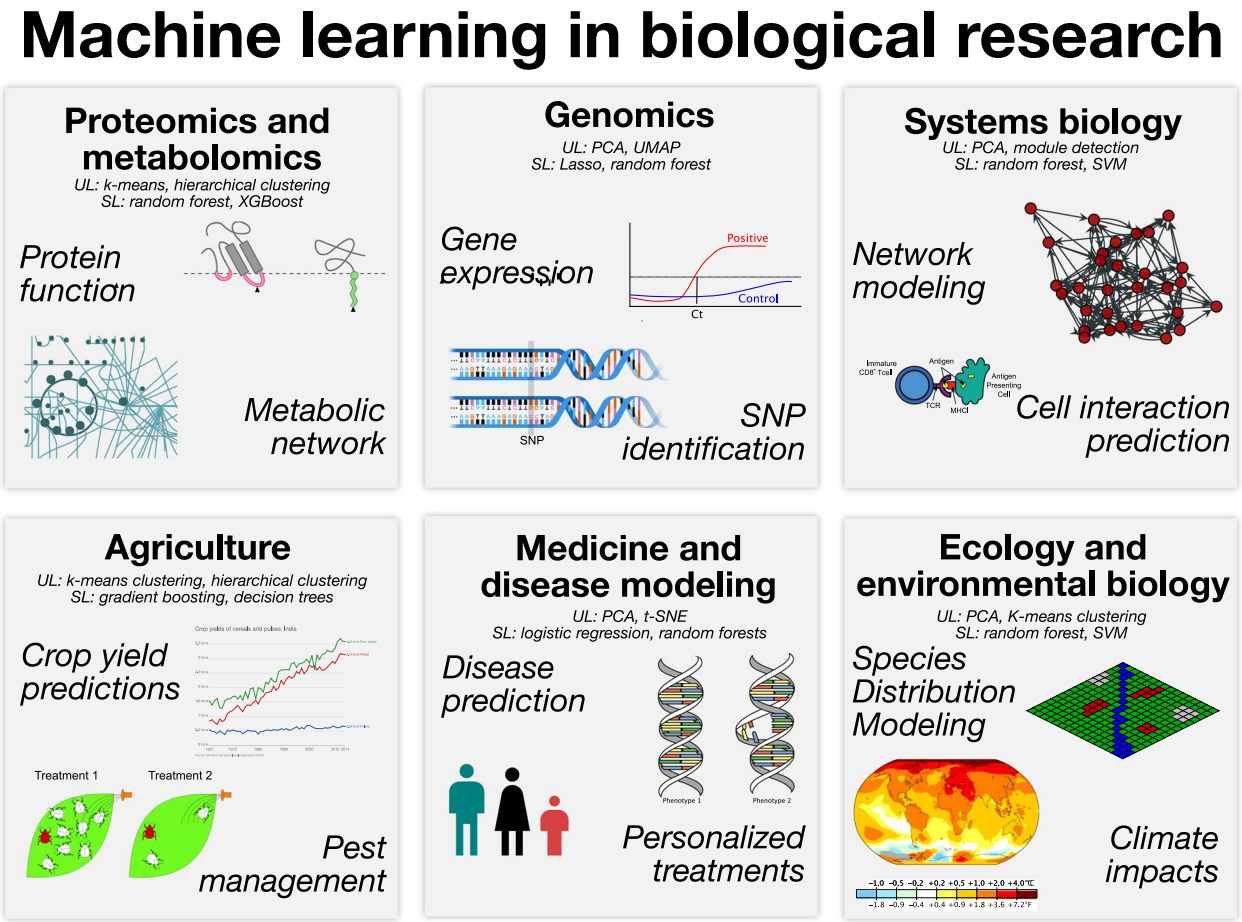

Fig. 1 Overview of key biological research domains where machine learning (ML) is actively applied. Each panel indicates representative tasks and commonly used ML approaches, including both unsupervised learning (UL) and supervised learning (SL). In genomics and proteomics, ML helps evaluate gene expression patterns, identify SNPs, and model protein function or metabolic networks. In systems biology, models support network modeling and cell interaction prediction while in agriculture, ML enables crop yield prediction and pest management. In medicine and disease modeling, models like logistic regression and random forest are used for disease prediction and personalized treatment strategies, while PCA and t-SNE assist in patient stratification. In ecology and environmental biology, classification tasks such as species distribution modeling often leverage random forests and SVMs, while PCA and clustering methods help explore change across gradients. All illustrations are adapted from Wikimedia Commons under appropriate open licenses

too complex and fails to generalize well, or underfitting, where a model is too simple to capture underlying trends. Generalization is therefore a key focus of ML. In practice, the goal of ML is to build models that effectively generalize from the training data to new data that follows the same distribution [5].

In addition to enabling direct prediction (e.g., forecasting, classification), ML can also help researchers make explanatory inferences from data. In inferential tasks, interpretability (i.e., the ability to determine which variables drive the model's decisions and how changes in input data affect outcomes) and the significance of variables often outweigh simple accuracy and performance metrics on data used to test the model (i.e., data held out during training). Additionally, ML encompasses a wide range of algorithms categorized into three main types: supervised learning, which relies on labeled data that has been annotated to impart context or other meaning; unsupervised learning, which seeks to identify the underlying structures of unlabeled data; and reinforcement learning, which involves models making decisions based on rewards received at each step of analysis through iterative trial-and-error processes [6]. Although our main focus is on supervised learning, we briefly touch upon particular scenarios when algorithms can work under either supervised or unsupervised learning frameworks (e.g., see support vector machines below). In general, understanding relationships and structures between datapoints within increasingly complex datasets is becoming ever more crucial and widespread in biological research, and such

questions highlight the need for selecting the appropriate specific ML approaches able to address the data [7].

ML has become integral to numerous tasks within biological research. For instance, ML has significantly enhanced precision, accuracy, and efficiency in predictive modeling, tackling biological questions at multiple scales. These range from prediction of molecular structures, to "omics"-level analysis, to pest identification and ecological forecasting [8–16]. These algorithms have enhanced the performance of genomic data analyses and influenced personalized medicine and genetic engineering across various domains [17, 18]. Nowadays, ML is useful in automating data processing, including high-throughput techniques like next-generation sequencing and high-content screening [19], reducing human error and even boosting the throughput and scalability of experiments. ML facilitates the integration of complex datasets, such as genomic, proteomic, and metabolomic data, allowing for comprehensive modeling of biological systems (e.g., [20]). These integrations have enabled researchers to incorporate more realism into understanding system-level interactions, particularly in fields like cancer biology and neurobiology [20, 21]. In ecological and environmental research, ML models are commonly used to predict environmental impacts on biodiversity and to guide conservation efforts amidst climate change and habitat loss [22, 23]. In genomics, the use of ML has become standard practice due to the complexity and sheer volume of data, aiding, for instance, gene expression profiling, single-nucleotide polymorphism (SNP) identification, and genomic sequencing. In the fields of proteomics and metabolomics, ML is central to tasks such as protein classification, function prediction, and metabolomic network analysis [24, 25]. Disease prediction and prevention heavily rely on ML frameworks, which are now standard in modeling disease outbreaks and progression. ML approaches are also critical in systems biology, where algorithms often help unravel cellular and intercellular interactions within cells and relations between organisms. Similarly, ML is used in modeling ecological dynamics, assessing climate-related impacts on biodiversity, and supporting conservation biology [26]. In agriculture, ML routines are used to predict crop yields, optimize resource use, and manage pest control effectively [27]. Most fields of biology which currently generate and analyze data are likely applying some type of ML to build models and predict patterns.

In this review, we highlight four machine learning algorithms that are widely adopted, thoroughly tested, and form the basis for more advanced techniques in the field (Fig. 1). We describe the algorithms, summarize their implementation in commonly used programming languages (R and Python), and outline recent applications in biology through a systematic literature review. Finally, we provide perspectives on the scalability of these tools to larger datasets, as well as future directions in the field, including applications of neural networks. We aim to provide an up-to-date perspective on the uses of ML in biology and establish connections between biological disciplines based on ML applications. Our review expands on the approach from Tarca et al. [28]. Yet, while Tarca et al. [28] provided fundamental perspectives on statistical and computational methods for analyzing high-throughput biological data, our review further incorporates recent algorithmic developments, examines cross-disciplinary applications, and emphasizes practical implementation strategies for contemporary biological datasets. Unlike prior reviews which were primarily focused on algorithmic theory or ML intersecting isolated biological domains, we provide practical, cross-disciplinary details that highlight methodological challenges, the relevance of model interpretability, and the need for a close integration of machine learning with domain-specific knowledge (see also [29–31]). We also highlight critical areas of integration where ML can significantly improve biological research, especially through interdisciplinary collaborations that bridge the gap between computational sciences and domain-specific knowledge.

Four key machine learning algorithms

We focus on reviewing recent research based on linear regression, random forest, gradient boosting machines, and support vector machines. For each of these algorithms, we provide an up-to-date introduction, followed by a technical description of the approach. Next, we outline two selected cases of recent and relatively impactful applications of each algorithm. Study cases were selected based on a systematic review, described in more detail in Additional File 1. We also address relevant challenges and considerations, including overfitting, data requirements, and interpretability in biological contexts. Finally, we discuss neural networks as a potentially relevant field for analogous questions where appropriate, as well as their impacts and future prospects in the context of biological research.

Reviewed algorithms were selected based on (1) widespread adoption across biological disciplines; (2) balance between predictive accuracy and interpretability; (3) complementary methodological approaches spanning linear, ensemble, and kernel-based methods; (4) accessible implementations in R and Python; and (5) their known scalability across diverse dataset sizes common in biological research. We also highlight that the emphasis on supervised learning reflects current

biological research priorities where labeled datasets (diseased/healthy, species classifications, functional annotations) are increasingly available through large-scale genomic, proteomic, and phenotypic studies.

For each algorithm, we describe two case studies from selected literature (see Additional File 1: Text S1 for details; Fig. 2). Briefly, we searched for biological papers using these algorithms, sorted them broadly by citation count, then selected the top two papers from a subset of papers that had been manually reviewed as demonstrating clear outcomes and implications of using the target algorithm. We compiled the resulting set of papers into a single spreadsheet (Additional File 2; see Additional Files 3–6 for algorithm-specific files). Papers published before 2020, papers without citations, and papers found for more than one model were removed (see Additional File 1, Text S2). We then sorted the subset of papers based on the number of citations at the time of retrieval. For each algorithm, we retained the top 50 papers based on citations. Next, we manually reviewed these papers to select case studies. We excluded review articles, book chapters, papers that did not explicitly use the method in question, papers that did not use it for biological research, or where machine learning methods did not inform the main results. We selected the top two papers from each set demonstrating clear outcomes and implications of using the target algorithm (Additional File 1, Text S3–S4) (Fig. 3).

Recent uses of machine learning in biology

Ordinary least squares regression

Overview

Ordinary least squares (OLS) is a statistical method that is used to estimate the parameters of a linear regression model [32]. OLS is sometimes also called a "best-fit line" (Fig. 4). This approach focuses on minimizing the sum of the squares of the residuals, which reflects differences between the observed values in the dataset and the values predicted by the model. In linear regression, the relationship between the dependent variable, y_i, and a set of independent variables, matrix x_i, is typically expressed as $y_i = \alpha + \beta x_i$. The coefficients β represent the parameters of the regression and summarize the influence of each input feature on the dependent variable. The term α is the intercept and captures the baseline value of y_i when all x_i values are zero. The sum of squared residuals, which is explicitly the target of OLS, is given by: $\sum_{i=1}^{n} \left(y_i - \alpha - \beta x_i\right)^2$. The least squares approach chooses α and β to minimize the residual sum of squares. Note that usage of the squared error is an analytical convenience but can over-emphasize outlier data points. Using some calculus, one can show that the minimizing values of α and β are:

$$\beta = \frac{\sum_{i=1}^{n} (x_i - \bar{x})\left(y_i - \bar{y}\right)}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

$$\alpha = \bar{y} - \beta \bar{x}$$

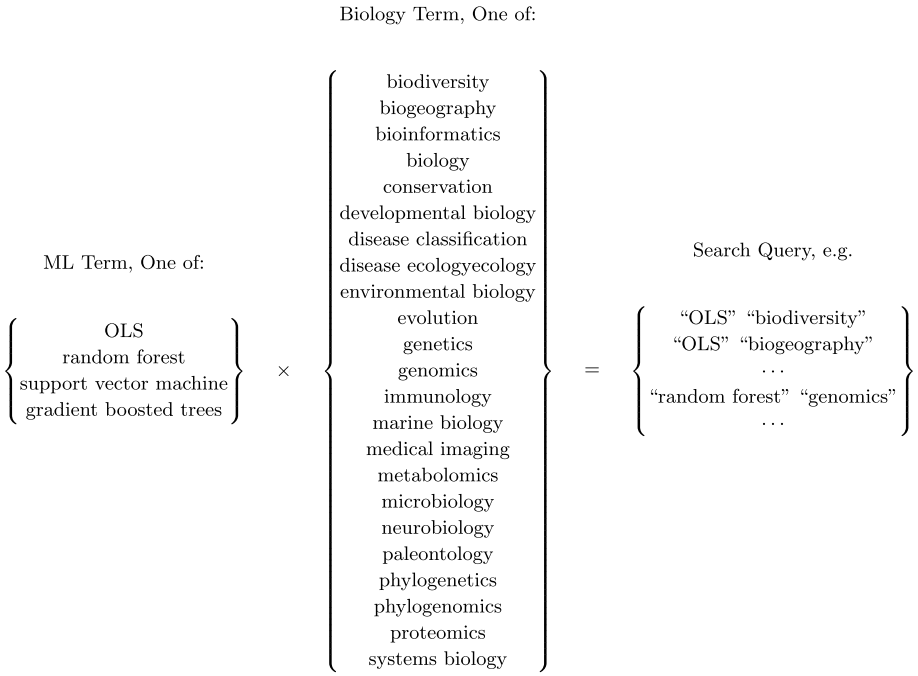

Fig. 2 Summary of the review approach used in this paper. We identified relevant recent papers based on a systematic literature review that used a predefined combination of terms describing the ML algorithm (first column) and the biological context of the paper (second column). The search query (third column) was generated as the full combination of all the terms in the first two columns

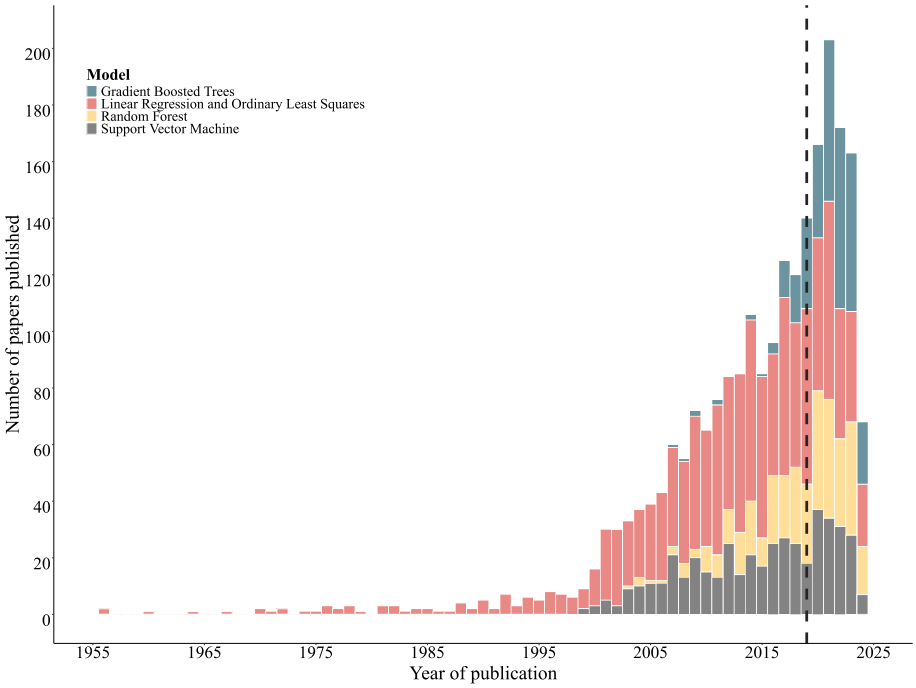

Fig. 3 Distribution of publications using the four machine learning algorithm examined in this study to test questions of biological relevance. We show results for four different algorithms including gradient boosted trees, linear regression models (two terms), random forest, and support vector machines. The vertical dashed line indicates the threshold used in this review to define recent studies (post-2020)

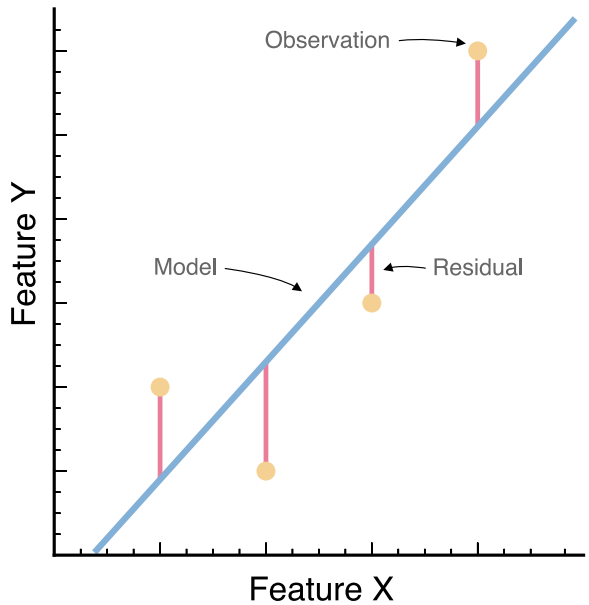

Fig. 4 Schematic representation of a linear regression model. We summarize the scatter for a hypothetical dataset (yellow circles), a given best-fitting model (blue line), the associated residual values (vertical red lines), in accordance to a response variable and a single predictor

where $\bar{x}$ refers to the arithmetic mean of a variable across the data set.

OLS works best when its underlying assumptions are followed, but there exist extensions for various situations. For example, by changing the squared error to an absolute error or even a median error, we can reduce the impact of outliers. Alternatively, if prior knowledge is available about the expected distribution of parameters, Bayesian regressions could provide a viable alternative to frequentist frameworks. Defining prior distributions on parameters is a form of "regularization," which typically helps models avoid overfitting and generalize better [32]. Likewise, if the dependent variable is a discrete class, one can modify OLS into a similar model such as logistic regression. Having been deployed in the sciences for decades, there are a plethora of OLS variants for many specific situations.

Some of the major advantages of OLS relate to its flexibility, interpretability, speed, and explanatory power. Specifically, because of the expected linear relationship between the response and independent variables, one can immediately infer the effect of changing a variable value on the prediction. Further, the underlying statistics enable calculating confidence intervals both on the predictions themselves as well as the parameter values (e.g., often this criteria is used to determine inclusion of an independent variable in a model). A key approach for

estimating uncertainty in parameter estimates is bootstrapping. Bootstrapping resamples the given data with replacement to create a new sample dataset of the same size. Then, parameters are re-estimated using the sample and are compared to the original parameter estimates by creating a distribution of a desired statistic (e.g., mean, median, confidence interval) for the target parameter. Finally, by requiring only elemental linear algebra, OLS is deterministic and fast. OLS often serves as a baseline against which other methods must compare.

Usage *in biological research*

Below, we outline two recent papers that explicitly use OLS to address questions on the intersection between ML and biology. First, Smith et al. [33] use a multiple linear regression, under a Bayesian framework (e.g., including prior distribution on regression parameters), to model the similarity between ecoregions as predicted by their geographical distance and environmental conditions. Ecoregions are large cohesive areas of land or water that are often described in terms of species assemblages, their ecological dynamics, or environmental conditions. In their paper, Smith et al. [33] use the Jaccard dissimilarity index (log-transformed) to capture the differences between ecoregions. This index is particularly used as the response variable in the examined models. Smith et al. [33] tests whether distance between ecoregions is explained by either (1) abiotic or (2) biotic factors. For their abiotic hypotheses, the independent variables were distance between regions, their mean homogeneity score, and principal components of environmental variables. For their biotic hypotheses, the independent variables were distance between regions, their mean homogeneity score, and either feeding guild or body size of terrestrial vertebrate taxa. Analyses also included the squared terms of predictors in the different models to account for a possible nonlinear relationship between their environmental predictors and distance between ecoregions. Analyses were conducted in Python using the "PyMC3" package [34]. Modifications to basic OLS include the Bayesian nature of the analysis (although uninformative priors were used). Significance of parameters was defined based on whether the relevant 95% credible intervals included 0, as is typical in statistical testing [32].

The second paper reviewed for OLS is Tao et al. [35]. In this study, the authors used linear regression models to compare estimates of phylogenetic divergence times between taxa as estimated by simple or complex models of molecular evolution. Complex models, represented in this study using GTR+Γ (general time-reversible), incorporate variable rates of nucleotide substitution. Simple models assume equal substitution rates and base frequencies. Simple and complex models were used to estimate divergence times across plant and animal clades. The explicit focus of the analyses was on node ages (i.e., branching times). Linear regression models were used to estimate the congruence between complex and simple models in terms of node age estimates. Time estimates were normalized by the sum of all node ages within each data set. The authors expected a linear pattern with low dispersion of points between the response and predictor, indicated by a slope close to 1 and high R^2 values (e.g., slope = 0.95, $R^2 = 0.99$), as a sign of high agreement between complex and simple models. We highlight that while linear regression was not the primary focus of the paper, it was particularly used to illustrate the similarity in divergence time estimates between complex models with many parameters, and simpler models, which are less computationally intensive, in the context of phylogenomic datasets.

Implementing linear regression models

Linear regression models can be fit in a number of different libraries implemented across multiple programming languages. We primarily focus on those in R or Python. Regression models can be fit using the 'stats' package [36]. A simple linear regression can be fit with the lm() function as indicated below. The training_data object is a table (e.g., data.frame) that includes a response variable (column y) and predictors (rest of the columns). The test_data object has the same structure of columns as training_data but was generated by splitting the full dataset into train (e.g., training_data, 70% of observations) and test sets (e.g., training_data, 30% of observations).

```r
library(stats)

# Fit the model with training data

ols.model <- lm(formula = y ~ ., data = training_data)

# Make predictions with the test data

preds <- predict(ols.model, newdata=test_data)
```

To run an Ordinary Least Squares (OLS) regression using Python, one can use the 'statsmodels' library [37] and assuming that training_data and test_data are pandas DataFrames:

```python
import statsmodels.api as sm

# Load training data

X = training_data[features_list]

y = training_data[target_feature]

# Create Model object

model = sm.OLS(y, X)

# Fit the model with training data

results = model.fit()

# Make predictions with the test data

predictions = results.predict(test_data)
```

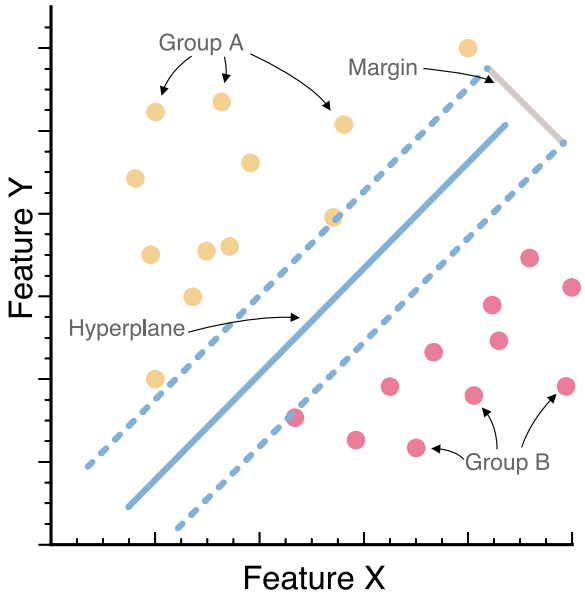

Fig. 5 Visual representation of support vector machines (SVMs) and its key elements. We present the relationships between two features in accordance to two different classes (group A in yellow, and group B in red). We show the hyperplane dividing the two groups, as well as the margin summarizing the overall division between classes

Support vector machines

Overview

Support vector machines (SVMs) are a set of supervised learning methods that are used in applications such as image classification, text classification, and various of bioinformatics routines. SVMs are often used for classification but can also be adapted for regression tasks. Similarly, although SVMs are generally fit for supervised learning, variations of SVMs can also be used under an unsupervised framework (e.g., one-class SVM). Before the 1980s, almost all learning methods learned linear decision surfaces, and the amount of samples in theoretical statistical studies was assumed to be large or infinite to simplify mathematical analyses. However, the size of empirical datasets is usually limited, and the relationships between features are almost never linear. In 1995, Vladimir N. Vapnik developed a novel approach and showed that SVMs work well with nonlinear and high-dimensional datasets at pattern recognition routines [5]. Based on the concept of similarity, SVMs use nonlinear "kernel" functions to transform the data to a higher dimension, enabling linear separation by finding optimal boundaries (i.e., hyperplanes) that form the best partition (i.e., decision boundary) between (1) classes and (2) support vectors or the data points that lie closest to the decision surface (or hyperplane) to maximize the margins between the classes (Fig. 5). SVMs are flexible in defining similarity measures and often generalize well to new data. With the advantage of global optimization and strong adaptability, SVMs have wide applications in areas like protein classification, and computer vision, among others.

SVMs are primarily focused on determining the hyperplane that optimally divides data into particular classes based on the maximum margin [5, 38]. The hyperplane is defined such that the minimum distance between data points in different groups (i.e., support vectors) is maximized. In the case of pairwise data, (x_i, y_i), where $(x_i \in R^n)$ are the feature vectors and $(y_i \in \{1, -1\})$ are the class labels, SVMs are focused on solving the following optimization problem:

$$\left[\min_{w,b} \frac{1}{2} \|w^2 + C \sum_{i=1}^{m} \xi_i\right]$$

subject to the constraints $\left(y_i(w \cdot x_i + b) \geq 1 - \xi_i\right)$ for every i. Note that $(\xi_i \geq 0)$ are slack variables that allow misclassification of either challenging or noisy points. Similarly, C is a regularization parameter that enables the control of the trade-off associated with achieving a high margin while reducing training error. The minimization itself, however, typically requires an iterative approximation as the non-linear kernel often precludes an analytical solution.

Kernels are the source of SVMs' intrinsic flexibility [39]. Kernels allow for operations in the input space

to be equivalent to operations in a higher dimensional feature space [40]. These operations based on kernels occur implicitly without the need of computing coordinates in the novel space [41]. For instance, assume that two populations of a given species inhabit distinct elevations and this is the key feature distinguishing them. However, elevation was not included as a feature in the dataset. Under SVMs, the use of a certain kernel on the features that were effectively collected in the dataset (e.g., latitude and temperature) could result in the indirect inclusion of elevation as a result of the expanded multivariate space (e.g., a proxy of elevation).

Mathematically, an SVM kernel function is the dot product of two vectors of higher dimensional space. Commonly used kernels functions polynomial kernel $k(x_i, x_j) = (\gamma x_i \cdot x_j + r)^d$, radial basis function (RBF) kernel $k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$, and sigmoid kernel $k(x_i, x_j) = \tanh(\gamma x_i \cdot x_j + r)$, where γ, r, and d are parameters that can be adjusted based on the data set.

Mathematically, an SVM kernel function generally involves the dot product of one data point with another, $\langle x_i, x_j \rangle = \sum_k x_{ik} x_{jk}$ where k indexes a given feature in the vector (e.g., temperature, latitude). These intermediate dot products can then feed into much more general non-linear functions such as the linear kernel or sigmoid kernel. The specific choice of kernel is beyond the scope of this review and is considered part of the larger model learning procedure (but see [32] for a detailed discussion). Because of this capability for handling nonlinearity, SVMs excel at domains where it is possible to draw a continuous "boundary" between data points of different classes. The nature of the kernel determines the shape capabilities of this boundary (e.g., a linear kernel will have boundaries which are linear in the independent variables).

When fitting SVMs, practitioners typically focus on fitting three critical parameters to optimize their models. First, the choice of the kernel type determines the transformation space of the input data. Each kernel type is suited for different types of data. For instance, the linear kernel is preferred for data that are linearly separable in the input space. The RBF kernel can handle more complex, nonlinear relationships. Second is tuning the regularization parameters, particularly the penalty parameter C and the kernel-specific parameter γ. These two parameters are essential for preventing overfitting and ensuring that the model generalizes well to new data. C controls the trade-off between achieving a low error on the training data and minimizing the model complexity for better generalization. The γ parameter defines how far the influence of a single training example reaches: low values indicate "far" and high values indicate "close." Third, defining the optimal value for the margin (i.e., the decision boundary) is crucial. A larger margin can increase the generalizability of the classifier. However, if the margin is set too wide, it might lead to misclassification of the training data, especially if the data are noisy or not well-separated.

Usage in biological research

We selected two case studies using SVMs. One paper focuses on detecting leaf disease using images [42] and the second on inferring taxonomic information of hosts based on viral genomes [43]. First, Das et al. [42] implemented a classifier to identify healthy and unhealthy tomato plants based on photos of their leaves. The authors focused on developing classifiers that could help improve the agricultural sector in India, ultimately enhancing the living standards of the rural population. For this study, the authors collected images from an existing database containing images of healthy and unhealthy tomato leaves ($n = 14{,}000$). They conducted image preprocessing steps and masking, including resizing and conversion to grayscale for further marking of target pixels. Color was extracted from RGB channels based on the masked images. These features (e.g., RGB channels, texture, contours) were then used to train and test random forest, logistic regression, and SVMs based on healthy and unhealthy classes. The training phase of the model was conducted on 60% of the images. The testing set, used for assessing model performance, comprised the remaining 40% of the observations. Das et al. [42] recovered a 25–30% higher accuracy for SVMs compared to random forest and logistic regression models. Based on these results, Das et al. [42] supported the deployment of SVM models for real-life applications in automatic disease detection at early stages.

Second, Young et al. [44] aimed to increase the available information on hosts for newly described viral genomes. The majority of newly discovered viruses lack taxonomic information for host species. The goal of this study was to identify genomic features of the virus that could be used to accurately predict the taxonomic information of the host. The key challenge was representing viral genomes in a format that made discriminative information available for ML procedures. For this study, sequences were retrieved from Virus-Host Database (VHDB) and RefSeq. Genomes were summarized as nucleotide sequences, amino acid sequences, physicochemical properties, and predicted PFam domains. From these representations, k-mer or domain extraction procedures were conducted to obtain a feature matrix. SVMs were trained on 80% of the data, with testing conducted on the remaining portion of the dataset (75% vs. 25% in alternative analyses). Phylogenetic information was accounted for in the analyses using a "holdout" method including an average

nucleotide identity filter. The SVM used a linear kernel, and performance was evaluated based on not just overall accuracy, but also using receiver operating characteristic (ROC) curves (equivalent to precision-recall curves and other approaches that simultaneously account for both false positive and false negative errors). The authors also combined different types of features derived from the same viral genomes and assessed their ability to predict host information. Based on their SVMs, Young et al. [44] found that all the analyzed feature sets were predictive of host taxonomy. However, combining feature sets had the potential to improve predictive accuracy further.

Implementation of SVMs

SVMs are implemented in a range of libraries both in Python and R. In R, the e1071 package [45], a wrapper of the LIBSVM C++ library, is standard for implementing SVMs.

```r
library(e1071)

# For regression

model <- svm(y ~ ., data = training_data, type = 'eps-regression', kernel = 'linear')

predictions <- predict(model, newdata = test_data)

# For classification

model <- svm(formula = class ~ ., data = training_data, type = 'C-classification', kernel = 'linear')

predictions <- predict(model, newdata = test_data)
```

Alternatively, excellent implementations of SVMs can be found in tidymodels, a package that offers a streamlined and modern framework for ML modeling within R [46]. The package includes functions such as svm_rbf() and svm_linear() that facilitate the application of SVMs with different kernel types in both regression and classification tasks.

In Python, the primary library for implementing SVMs is scikit-learn [47]. This versatile library implements functions to fit and analyze SVMs including SVC (Support Vector Classification), SVR (Support Vector Regression), and LinearSVC, an implementation that supports linear and non-linear SVMs. Additionally, the BioPython toolkit and related libraries offers closely integrated SVM-related implementations [48]; https://biopython.org/wiki/Scriptcentral.

```python
from sklearn import svm

# For regression

model = svm.SVR(kernel='linear')

# Fit the model with training data

model.fit(X, y)

# Make predictions with the test data

predictions = model.predict(test_data)

# For classification

model = svm.SVC(kernel='linear')

# Fit the model with training data

model.fit(X, y)

# Make predictions with the test data

predictions = model.predict(test_data)
```

Random forest

Overview

Random forest (RF) is a machine learning technique that has gained widespread popularity among researchers and practitioners due to its versatility and effectiveness, particularly in prediction tasks [49, 50]. This approach builds on an ensemble of decision trees that, besides predictive and inferential tasks, enable feature selection procedures as part of analyses and explicitly model interactions between variables [49, 51–53]. RF belongs to the ensemble learning family, a framework that combines multiple individual models to improve overall predictive performance. The "forest" in random forest is composed of decision trees (Fig. 6). A decision tree resembles a flowchart structure where each internal node represents a threshold or definition based on a particular feature. Branches represent decision rules and each leaf node represents an outcome. Decision trees are simple and easily interpretable yet effective models for classification and regression procedures [32].

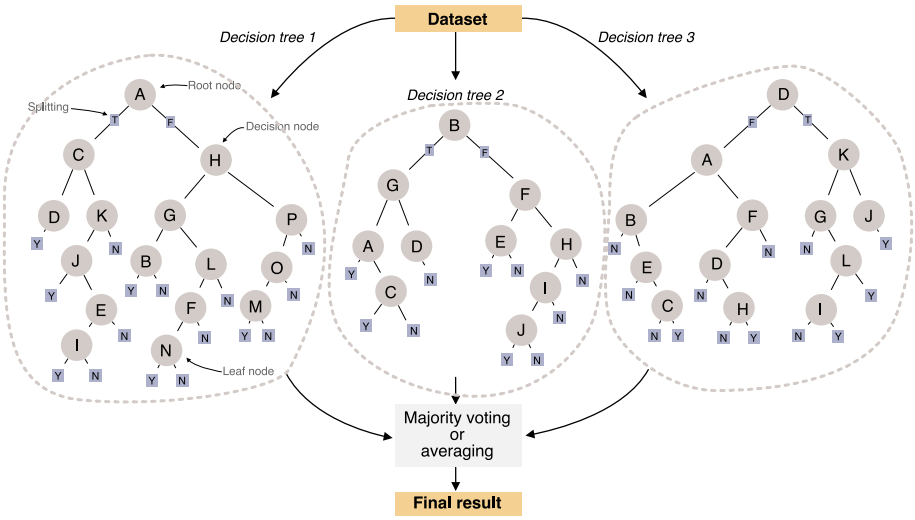

Fig. 6 Illustration of a random forest algorithm based on three decision trees and a number of features used to train the model (letters in circles). We show different decision trees that are trained independently from an initial dataset. Results across decision trees are summarized using either majority voting or averaging

There are at least six aspects that are critical to understand the structure, fit, and performance of RF algorithms. First, RF employs a sampling technique called bagging. This approach involves training each decision tree on a random subset of the training data (sampled with replacement, so a data point can occur multiple times for the same tree) that ultimately reduces overfitting by introducing diversity among trees. Second, each decision tree in the RF is constructed using a subset of features selected randomly at each node. This randomness ensures that trees are less correlated with each other, leading to a more robust model. Third, one of the hyperparameters of RF is the number of trees in the forest. Typically, increasing the number of trees improves performance while increasing computational cost. Finding the optimal number of trees often involves cross-validation techniques (i.e., trying many different values on subsets of the data while scoring on held-back data). Fourth, RF provides a measure of feature importance, which indicates the contribution of each feature in predicting the target variable. This information can be useful for feature selection and understanding the underlying data. Fifth, training each decision tree in RF is independent of the others, making it highly parallelizable. Many implementations of RF leverage parallel computing to speed up the training process, especially when dealing with large datasets. Sixth and finally, RF has several hyperparameters such as the number of features to consider at each split, maximum depth of the trees, minimum samples per leaf, among others to offer additional flexibility. Grid search or randomized search techniques can be used to find the optimal combination of hyperparameters [32]. These key aspects all contribute to the ongoing effectiveness and popularity of RF.

Note that there are a variety of tree-based ensembles that are structurally similar to random forests. For instance, we will describe gradient boosted trees later in this review. Bayesian additive regression trees are also a popular approach that also fall within the tree-based ensemble framework [54]. Each method, however, has a different training procedure. Just as Bayesian linear regression has the same structure as OLS, tree ensembles can come in many forms with different trade-offs.

Usage in biological research

We explore the use of random forest in two case studies. First, Fabris et al. [52] used random forest to discern loci underlying both discrete and quantitative traits, particularly when studying wild or non-model organisms. RF is becoming increasingly used in ecological and population genetics because, unlike traditional methods, it can efficiently analyze thousands of loci simultaneously and account for non-linear interactions. The authors described how to prepare data for RF, including initial data exploration, the identification of important features, and possible confounding factors. They then provided guidance on the initiation of RF and the optimization of the algorithm parameters for classification and regression. Finally, they summarize methods for interpreting the results of RF and identifying trait-associated or predictor loci. Second, Brieuc et al. [51] focused on looking at how RF can be used effectively in studies focused on genotype–phenotype associations, particularly in non-model organisms. This study, structured as

an introductory guide to the intersection between RF and ecological and evolutionary genomics, discusses fundamental approaches to carefully fit, analyze, assess performance, and understand results of RF-based approaches.

Implementation of random forest models

We present implementations of random forests using Python and R. In R, random forests can be fit with a given dataset using the randomForest package [55] as follows:

```r
library(randomForest)

# Initialize the random forest model

rf_model <- randomForest(x = X_train, y = y_train, ntree = 100, importance = TRUE, classwt = "balanced")

# Make predictions on the test data

rf_pred <- predict(rf_model, X_test)
```

Random forests can also be implemented in Python using the scikit-learn library [47]:

```python
from sklearn.ensemble import RandomForestClassifier

# Initialize the Random Forest model

rf_model = RandomForestClassifier(max_depth=2, random_state=0)

# Train the model

rf_model.fit(X_train, y_train)

# Make predictions on the test data

rf_pred = rf_model.predict(X_test)
```

Note that where SVMs have a few choices of kernel function and possibly one or two other parameters, RF has a plethora of so-called hyperparameters. One can vary the number of trees, features considered in splitting, the required purity for a node to stop being split, and more. Implementations of RF generally expose these parameters at model initialization, implying that trying different options via cross-validation or similar techniques is recommended. Methods like grid search, random search, or Bayesian optimization are common techniques for systematically optimizing these hyperparameters. Specifically, grid search exhaustively searches through a specified range of parameter values, while random search randomly samples from a distribution of parameter values. Bayesian optimization uses probabilistic models to focus the search on promising regions of the parameter space. Cross-validation is typically used to evaluate the performance of different parameter configurations and select the best one. Additionally, Out-Of-Bag (OOB) performance metrics are also commonly used in RF. OOB refers to the data points excluded from a bootstrap sample when training individual trees in an ensemble model. These data points are used as a built-in validation set to estimate the predictive power of each tree within the forest [11, 33].

Gradient boosting

Overview

Gradient boosted models (GBMs) can be understood by extending our prior explanation of random forest. Where random forest [49] creates an ensemble of trees through bagging, gradient boosting develops each component model (i.e., individual decision tree) of the ensemble one after the other [32]. This iterative procedure is generally called boosting (Fig. 7). Specifically, let $f_{m-1}(x_i)$ be the boosted model's prediction after $m-1$ components have been added. Under this boosting, we seek the next iteration, $f_m(x_i) = f_{m-1}(x_i) + \Gamma_m g_m(x_i)$ (i.e., one has generated two successive models via GBM and seek to refine it by adding a third component to the ensemble). For example, one could fix $\Gamma_m = 1$ and fit g_m to minimize the residual loss, $L\big(y_i - f_{m-1}(x_i), g_m(x_i)\big)$. That is, each new component attempts to correct errors of the previous model. The way to determine g_m and γ_m depends on the exact nature of the boosting. One subtype of boosting is called gradient boosted models [56] or GBMs. This approach fits g_m to minimize the loss on the negative gradient, $-\frac{\partial L(y_i)}{\partial f_{m-1}}$. Then one finds the weight, Γ_m to minimize the overall loss, $L\big(y_i, f_{m-1}(x_i) + \Gamma_m g_m(x_i)\big)$. The gradient helps direct the next model more carefully than generic boosting.

The exact capabilities of GBMs depend strongly on the underlying model type within the ensemble. For example, gradient boosted trees (GBTs) are structurally identical to random forests, and they are often applicable to similar problems. Because GBMs use a gradient, however, they can take advantage of a continuous loss function to speed-up model convergence. Conversely,

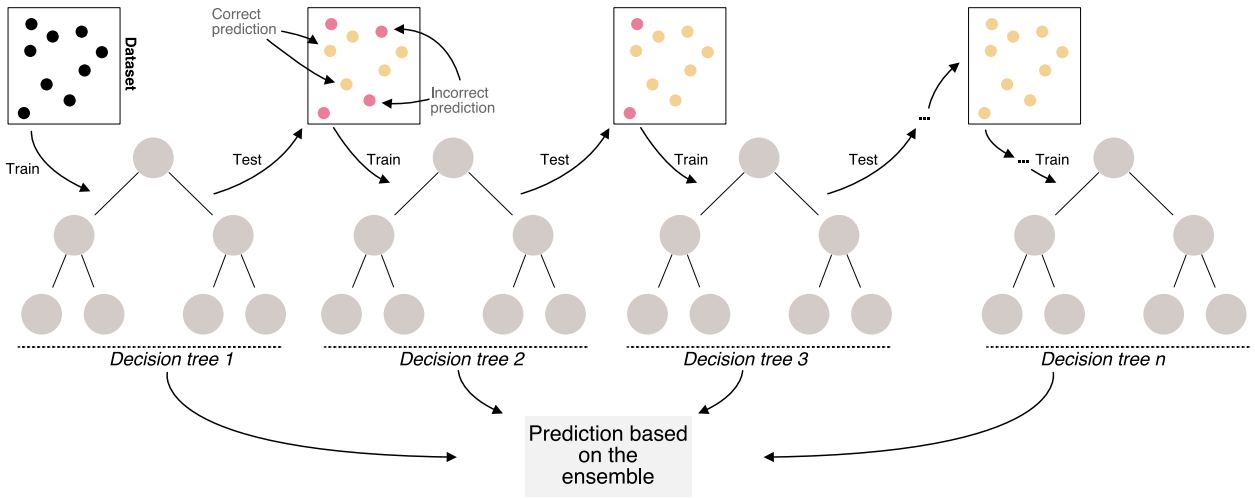

Fig. 7 Diagram showing the temporal patterns in training for Gradient Boosting Trees. We show an initial dataset used for training of a decision tree. Subsequent iterations of the boosting sequence are focused on enhancing model performance by reducing error rates. We show correct model-based predictions in yellow. Red circles summarize model errors. At the end, model-based predictions are performed on the tree ensemble

problems with discontinuous losses may be less suitable for GBMs.

Usage in biological research

We selected two case studies that summarize the use of gradient boosting in the context of biological research. First, Zhang et al. [57] built a predictive model for bioluminscent proteins (BLPs) via sequence-derived features for identification. BLPs are valuable for both industry and research. In this study, the authors used XGBoost (eXtreme Gradient Boosting), an ensemble learning algorithm based on gradient boosted trees [58]. XGBoost is well known for its highly flexible and scalable tree structure enhancement model and its reduction of computational time and memory for training large-scale data. All of these features were used to specifically improve on methods and tools previously used for the prediction of BLPs. First, a previously constructed comprehensive dataset consisting of BLP sequences and non-BLP sequences composed from bacteria, eukaryote and archaea, was collected from UniProt to be used as training and predictive data. In order to avoid homology bias, the data was first cleaned by using BLASTClust [59]. The variety of features which make these sequences identifiable and which was characterized by previous work, was further encoded via various methods (i.e., natural vector, composition/transition/distribution, g-gap dipeptide composition, and pseudo amino acid composition), mainly so that the data could be processed through ML algorithms like XGBoost. The dataset was then used to develop the prediction model by finding the highest area under the curve (AUC) values (again, a measure of overall accuracy and the trade-off against false alarms) correlated to specific encoded features for which each of the species in the set could be optimized for prediction. Performance was then measured by testing the data using different combinations of encoding features along with XGBoost's internal statistical analyses for cross-validation. Results indicate strong predictions for species-specific trained data sets and overall more accurate results when compared to other predictive algorithms (e.g., decision trees, random forests, and AdaBoost).

Second, Yaşar et al. [43] used multiple ML predictive algorithms (deep neural network, random forest, and GBT) to classify three COVID-19 positive patient groups (mild, severe, and critical). This study also aimed to generate a control group by using blood protein profiling as predictive indicators. The team obtained a dataset that included age, gender, and 368 proteins obtained from blood protein profiling. The number of people in each severity group and control group varied. In order for the data to be successfully used by the algorithms, it first had to be standardized in multiple ways to account for inequalities (e.g., missing values, unbalanced sample size). After the data was cleaned, the ML algorithms were trained to make predictions about disease severity. The GBT algorithm was worked in as noted,a prediction function was iteratively refined. Residuals computed from the difference between predictions and actual data subsequently informed the next target in GBT, creating new and more accurate residuals which were used as training data as further iterations occurred. Using ML predictive algorithms, the authors identified 10 proteins associated with COVID-19 severity that could be used

as bio-markers, with two of them (IL6 and LILRB4) supporting results of previous proteomic-based work. GBT achieved the best prediction of disease severity based on available proteins compared to the rest of the tested algorithms according to the results of various metrics (i.e., accuracy, sensitivity, specificity, precision, classification error, and others).

Implementation of Gradient Boosting

Libraries to implement GBMs exist in many programming languages. For this review, we offer two code snippets, one in R and one in Python, to demonstrate one way to deploy this method in a simple dataset, split into train and test sets. First, GBMs can be fit in R using the gbm package [60] as follows,

```r
library(gbm)
## Build a predictive model for regression or classification
# There are several tuning parameters in the gbm package that this example code leaves out for simplicity. Users can specify the number of trees, interaction depth, and cross-validation folds, among other parameters, to tune the model.
# Train the model
model<- gbm(y ~ ., data = training_data)
# Using the model, make predictions on the test data
predictions<- predict(model, test_data)
```

Alternatively the xgboost library in Python can be used to fit a Gradient Boosted Trees model with XGBoost [58]

```python
import xgboost as xgb
# For regression
# Initialize the model
reg = xgb.XGBRegressor(n_estimators=10)
# Train the model
reg.fit(X_train, y_train)
# Make predictions on the test data
predictions = reg.predict(X_test)
#For classification
# Initialize the model
clf = xgb.XGBClassifier(n_estimators=10)
# Train the model
clf.fit(X_train, y_train)
# Make predictions on the test data
predictions = clf.predict(X_test)
```

Challenges of ML-based inference in biology

Despite the flexibility and potential of machine learning models in biological research demonstrated through the examined case studies, ML as an analytical framework still suffers from various constraints that limit its performance and widespread use within disciplines. In this section, we briefly review the major pitfalls associated with using a machine learning framework to address questions of biological relevance. We focus on the models examined in this paper, detailing the limitations of machine learning in biology and examining the potential for future advancements. It is essential to critically evaluate limitations to best leverage machine learning in biological research. Furthermore, the applicability and reliability of these models in the field can directly be understood from exploring innovative solutions to these challenges.

A common challenge when applying machine learning as an analytical framework to answer biological questions relates to selecting the most effective technical implementation of a given algorithm (see also [61]). For instance, although this paper presents only a single SVM framework, there exists a wide range of alternative approaches for analyzing SVMs (e.g., svmSomatic, GraDe-SVM, MD-SVM). The same situation applies to GBMs (e.g., LightGBM, XGBoost, CatBoost), linear regressions (e.g., ridge regression, lasso regression, Bayesian linear regression), random forests (e.g., extra trees, oblique random forests, rotation trees), and most—if not all—algorithms within a machine learning framework. At this point, users will likely face several important questions. First, which of the available alternatives should be used? Second, do different implementations affect the results? We encourage less experienced users to evaluate model performance using the simplest version of the implementation before exploring other alternatives (e.g., [62, 63]). In some cases, explicit constraints or specific data structures justify the use of question-specific models (e.g., TF-Boosted Trees, a TensorFlow implementation for structured data problems). In such cases, users should prioritize more specialized approaches in accordance with their knowledge of the data, the question, and the algorithm. Trying multiple algorithms is an option—and often common practice—but more importantly, users should critically assess why fitting a particular method is advantageous as an a priori step before examining its performance.

Another critical aspect to machine learning, closely linked to both model fitting and performance assessment, is data visualization (e.g., visual analytics; [64–66]). Despite being oftentimes overlooked in favor of model-focused paradigms when machine learning is introduced, data visualizations are integral to understanding both the process and results of machine learning. In many cases, model performance can be evaluated directly by visualizing relevant features, particularly when guided by domain knowledge [67]. For instance, in classification tasks, visualizations such as logistic regression nomograms can provide a clear understanding of how feature values influence a model's predictions. These visual representations allow researchers to distinguish patterns that may not be immediately apparent in raw data or numerical outputs. Furthermore, assessing model performance or identifying violations of assumptions often involve creating plots such as residual plots (e.g., regression) or ROC curves (e.g., classification, [62]). These types of visual tools can help reveal the accuracy, precision, and limitations of a model. Interpretability is inherently tied to data visualization, as it transforms complex,

often abstract model behavior into a format that can be easily understood. We highlight that the true success of a machine learning model lies not just on its ability to predict a given outcome accurately, but in its capacity to explain that prediction in an accessible manner [68, 69]. Models may capture patterns—some visible, other invisible—that, when visualized appropriately, offer valuable insights into underlying biological phenomena [58]. Data visualizations, therefore, are not only a tool for model validation but also for conveying the nuances of model performance. Visualizing data, models, and their performance make the entire process more transparent and interpretable.

In the following paragraphs, we will discuss the intersectionality between each of the four algorithms outlined in this review, in the context of biological research. Although OLS is a basic and widely used method in machine learning, it relies on multiple assumptions that are often not met by many datasets (e.g., constant error variance and independent data points). For instance, in a dataset including phylogenetically related organisms, OLS may fail to reveal relationships present within but not across clades. Conversely, OLS might indicate relationships across clades that do not exist once phylogeny is accounted for. In such cases, phylogenetic generalized least squares (PGLS) explicitly incorporates information on relationships between terminals through a variance–covariance matrix and may be a better approach [70]. This example shows that the structure of the dataset is highly relevant for ultimately selecting the appropriate analytical tools. Even though OLS has been shown to be robust to some violations of its assumptions (e.g., [71, 72]), there should be a particular focus on reviewing the appropriateness of the method in accordance with the focal question. Although linear regression models are straightforward to implement, researchers should consider whether their aims are predictive or causal. For causal inference, careful selection of predictors is crucial. Tools such as directed acyclic graphs should be used to make explicit the hypothesized relationships between variables [73]. In some scenarios, alternative modeling approaches may better accommodate the complexities of biological data. A flexible and iterative framework of model selection and evaluation can enhance the robustness of the findings.

In the context of SVMs, preprocessing and interpretability are critical aspects of this approach. First, preprocessing of datasets analyzed using SVMs generally involves several crucial steps. Normalization is critical as SVMs are not scale-invariant, requiring features to be scaled to have zero mean and unit variance to ensure the model does not bias toward attributes with higher magnitude values. Imputation, the process of replacing missing data with an estimated values, is necessary for SVMs to be successful. Missing values are common in biological datasets. Balancing is particularly important in classification tasks where class distributions are uneven, as unbalanced datasets can lead to biased models that overpredict the majority class. Thus, developing new or systematically enhancing the existing preprocessing techniques can significantly improve the performance of SVMs in biological applications. Second, relative to simpler models, the interpretability of SVMs is generally challenging. Understanding the estimated weights is the primary focus, but the use of kernels often leads to the exploration of novel multivariate spaces, making it difficult to interpret results in the context of the biological question. Despite these challenges, we reviewed two case studies in which SVMs were successfully used to answer questions in the field mostly based on model performance (not on variable importance). Enhancing the interpretability of SVMs, either through innovative visualization approaches or explanation methods, could make these models more accessible to researchers.

Key aspects to account for when analyzing random forest models include overfitting, data requirements, model complexity, validation, and generalization. First, while RF is more robust against overfitting compared to individual decision trees, overfitting can still occur, especially with noisy or high-dimensional biological data. Careful tuning of hyperparameters such as the number of trees, maximum depth, and minimum samples per leaf is necessary to mitigate overfitting and ensure generalization to unseen data. Another approach to parameter tuning involves incorporating domain-specific knowledge into the model tuning process. Depending on the dataset used with RF, the model is more sensitive to changes in different hyperparameters. For example, Huang and Boutros [74] found that in a next-generation sequencing quality-control dataset with a low p/n ratio (variables/samples), the m_{try} parameter (number of variables to sample) had a significant impact on the resulting model, while the number of trees and sample size did not. In an mRNA abundance dataset with a high p/n ratio, all three parameters (m_{try}, number of trees, and sample size) had a significant impact on the resulting model [74]. Employing domain knowledge allows researchers to focus on tuning specific hyperparameters that have the greatest impact on the model. This external knowledge can enhance the applicability of RFs into specific datasets. Second, RF performs well with large datasets. However, biological datasets often pose unique challenges such as imbalanced class distributions, missing values, and high dimensionality. Pre-processing steps like feature selection, imputation, and data balancing are thus crucial to optimize model performance and prevent biases. Third, RF can handle

complex relationships between genetic variables and phenotypic traits, but it may not capture subtle interactions or continuous relationships present in biological systems. Model interpretation and validation techniques, such as permutation importance and partial dependence plots (i.e., visualizations used to examine how model predictions perform as (i) one or multiple inputs change while (ii) the rest remains constant), help elucidate the relationships between genetic predictors and ecological or evolutionary outcomes. Fourth, assessing the performance and generalization of RF models across different populations, species, or environmental conditions is essential in biological studies. Cross-validation techniques, independent validation datasets, and robustness testing help ensure the reliability and applicability of random forest models in diverse ecological and evolutionary contexts.

Finally, GBMs have a powerful predictive performance. However, this outstanding performance comes with a set of challenges that often needs to be addressed in order to maximize their utility in biological research. First, GBMs are prone to overfitting, especially with noisy or high-dimensional biological data. Appropriately tuning parameters such as learning rate, number of trees, and tree depth is thus crucial to achieve an optimal performance across different datasets. Second, the computational complexity of GBMs can be high, often requiring significant computational resources and time. Resource-related limitations are often one of the reasons why GBMs are limited to smaller datasets or procedures of low complexity. Third, the interpretability of GBMs is generally lower compared to simpler models [75, 76]. In this context, the ensemble of decision trees can obscure the understanding of feature importance and interactions (see also RF). However, techniques like SHapley Additive exPlanations (SHAP) values or partial dependence plots are often used to provide details of the model's decision-making process [77, 78]. SHAP values, for example, measure the contribution of each feature to individual predictions [77]. SHAP thus allows for a more granular understanding of model behavior in the context of the relevant task. Partial dependence plots, on the other hand, show the relationship between a selected feature and the predicted outcome. These plots show how changes in feature values impact model predictions. Future developments in explainable AI may enhance the interpretability of GBMs, potentially making these models more user-friendly for biological researchers. Fourth, handling missing data and imbalanced class distributions in biological datasets requires robust preprocessing strategies that ensure GBMs to produce unbiased predictions [79]. Missing data and class imbalance are, however, common in biological datasets. Lastly, rigorous cross-validation, independent test sets, and domain-specific adjustments to the model are often needed to ensure the generalization and validation of GBMs across different datasets (e.g., varying species, environmental conditions, or population structures).

Outlook: deep learning in biology

Although the main focus of this review has been on foundational machine learning and its intersection with biological fields, deep learning (DL) methods (i.e., neural networks) have been gaining strong relevance in multiple subdisciplines due to their flexibility. The basic DL framework is inspired by neuronal connections. Nodes in the graph, often referred to as neurons, are connected in layers of variable length. The first layer of neurons processes the input features and the final layer computes the output, using the previous layer as input. In its simplest form, neurons in each layer share connections to neurons in adjacent layers. The model is generally parameterized by a learning algorithm that propagates feedback to the internal weights attached to neurons from the training data through these connections. This results in a parameter-rich model capable of decomposing meaningful signals from highly complex feature sets [80].

Numerous variants of neural networks and learning models have emerged over the years. Deep learning techniques that excelled at image recognition were being applied in biological sciences going as far back as the late 1990s (e.g., [81–83]). Early applications were limited to segmentation of medical imagery [3], disease recognition [84], and diagnosis [85]. Classification models have demonstrated impressive accuracy on test data [86–88] but await refinement before adoption into widespread clinical use [89]. Other examples of early explorations can be found in drug discovery [90], virtual screening [91], and functional genomics [90, 92]. Deep learning methods have amassed greater traction from the biological science community in recent years. For instance, following rapid advances in high-throughput sequencing and the popularization of resequencing-based studies [93], deep learning techniques have made notable strides in variant detection for molecular pathologies [94, 95]. First in CASP13 in 2018 [96] and later again CASP14 in 2020 [97], Google DeepMind's AlphaFold triumphed over competitors demonstrating resounding classification performance on protein folding patterns. Today, advancing medical technologies, the promise of personalized medicine and an abundance of DNA sequence data at the population scale have placed biological and life sciences at the forefront of explorative research in deep learning [98].

Recent breakthroughs in deep learning (e.g., transformers and large language models such as ChatGPT and Gemini) have initiated a new age of generative AI

(Artificial Intelligence; [99]). Both public and commercial attention to this field of science has never been greater. There has been a cumulative increase in data center and research funding from government and private sectors [100–102]. Dramatically improved parallel computing infrastructure made possible by fabrication of as low as 3 nm process semiconductors [103] and high-speed interconnect between compute units [104] pave the way for greater advancements. These generative approaches show some promise, especially in their breadth of apparent knowledge, but they are prone to "hallucination," sometimes fabricating plausible but incorrect or untested concepts as a result of their probabilistic nature and weakly curated data sets [105]. As the subfield of artificial intelligence and deep learning keeps gaining momentum, reintroduction of proven methods and the development of new algorithms tailored to the needs of biological sciences is essential.

Our review presents DL and ML as viable approaches for answering questions in biological disciplines. The choice between traditional machine learning and deep learning hinges on the nuanced trade-offs among data availability, computational resources, model interpretability, accuracy, and scalability, demanding careful consideration tailored to the specific requirements and constraints of each biological research question. Below, we present three key aspects that highlight the pragmatic difference between traditional ML and DL. First, the relevance of DL is primarily evident in applications where large semi-structured data sets are available (e.g., images of cells instead of tables). Relative to traditional ML algorithms, large DL models have demonstrated an unprecedented ability of processing language and visual data and demonstrated crude deductive capabilities [106, 107]. Second, when data is limited, traditional machine learning methods often outperform DL models, which require extensive datasets for optimal performance. The drawback of DL models is their requisite dependence on an abundance of high quality and often manually curated training data [108]. Second, the training time required for DL models can significantly exceed that of traditional methods, thus influencing decisions based on computational resources and time constraints. Recent advances in parallel computing hardware and community-accelerated software development have enabled the construction of gargantuan models comprising billions of parameters, though training speed remains an issue for specialized analyses [109]. Third, method complexity and interpretability remain critical considerations,while DL techniques often achieve superior accuracy in complex pattern recognition tasks, this accuracy frequently comes at the cost of reduced interpretability and transparency. Conversely, foundational machine learning models such as linear regressions or decision trees offer greater interpretability, making them preferable in scenarios where explainability and transparency are crucial, even if they may occasionally sacrifice a degree of predictive accuracy.

Conclusions

As shown by the case studies in biogeography [33], agriculture [42], virology [44], clinical diagnostics [43], and others, machine learning is expected to keep playing an increasingly vital role in biological research in the coming years. The adaptability of machine learning frameworks enables researchers to tailor models to their specific datasets, often yielding predictions that are more reliable than other traditional methods. However, challenges such as parameter tuning, preprocessing, and the complexity of biological data must be carefully addressed to prevent overfitting and ensure model interpretability. We highlight that the incorporation of domain-specific knowledge into the model tuning process can be crucial for overcoming issues related to uninterpretable results (e.g., [26]). To this end, collaborations between data scientists and domain experts across different fields can significantly enhance the validity of machine learning applications [110]. Thus, fostering interdisciplinary research by integrating domain-specific expertise into machine learning projects help account for model-specific tradeoffs and further enhance the interpretability of results.

Looking ahead, the future of ML in biology will likely leverage deep learning techniques, including transformers and large language models, which have already shown promise in predicting [111] and designing [112] protein structures and understanding single-cell interactions [113], among others. Similar to how ML and big data in general affect society, the deployment of these advanced models raises critical ethical considerations, such as data privacy and bias in training data. We highlight the Findable, Accessible, Interoperable, and Reusable (FAIR) principles, a framework that ensures data accessibility with its ethical use, and supports transparency and community engagement in data sharing [114, 115].

Abbreviations

AI	Artificial intelligence
AUC	Area under curve
BLP	Bioluminescent protein
DL	Deep learning
FAIR	Findable, Accessible, Interoperable, and Reusable
GBM	Gradient boosted models
GBT	Gradient boosted tree
GTR	General time-reversible
ML	Machine learning
OLS	Ordinary least squares
OOB	Out-Of-Bag
PGLS	Phylogenetic generalized least squares
RBF	Radial basis function
RF	Random forest

RGB	Red Green Blue
ROC	Receiver operating characteristic curve
SHAP	SHapley Additive exPlanations
SNP	Single-nucleotide polymorphism
SVC	Support vector classification
SVM	Support vector machine
SVR	Support vector regression
TF	TensorFlow
VHDB	Virus Host DataBase
XGBoost	EXtreme Gradient Boosting

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12915-025-02424-3.

Additional file 1. Methods followed for the systematic literature review. Text S1. Details on systematic searches. Text S2. Procedures used to subset the relevant studies for each of the selected algorithms. Text S3. Details on the information extracted for each of the analyzed papers. Text S4. Details that were highlighted in the implementation details associated to each algorithm.

Additional file 2. List of all papers retrieved for our analyses.

Additional file 3. OLS search results and coded papers.

Additional file 4. SVM results and coded papers.

Additional file 5. Random Forest search results and coded papers.

Additional file 6. GBM search results and coded papers.

Acknowledgements
We thank members of the Data Diversity Lab at the University of Arizona (Spring 2025) for discussions. We thank three anonymous reviewers for their comments and constructive feedback on the manuscript.

Authors' contributions
Md Nafis Ul Alam: Writing—Original draft preparation & Reviewing and Editing. Kiran Basava: Data curation, Writing—Original draft preparation& Reviewing and Editing. Ani Chitransh: Data curation, Writing—Original draft preparation. HM Abdul Fattah: Data curation, Writing—Original draft preparation. Hector D. Garcia-Verdugo: Data curation, Writing—Original draft preparation & Reviewing and Editing. Shih-Hsuan Lo: Data curation, Writing—Original draft preparation. Tanisha Lohchab: Data curation, Writing—Original draft preparation. Kristen M. Martinet: Writing—Reviewing and Editing. Cristian Román-Palacios: Data visualization, Writing—Original draft preparation & Reviewing and Editing, Conceptualization, Methodology, Supervision. Jhan Carlos Salazar: Data visualization, Writing—Original draft preparation & Reviewing and Editing. Danielle Van Boxel: Data curation, Methodology, Writing—Original draft preparation & Reviewing and Editing, Methodology.

Funding
Not applicable.

Data availability
All data generated or analyzed during this study are included in this published article, its supplementary information files and publicly available repositories. Data S1–S5 can be found in the supplementary files to this paper. Data and relevant code used to conduct searches, curate datasets, and plot figures can be found in the following GitHub repository: https://github.com/datadiversitylab/MLBioReview and Zenodo at https://doi.org/10.5281/zenodo.17089842.

Declarations

Competing interests
The authors declare no competing interests.

Received: 4 February 2025 Accepted: 22 September 2025
Published online: 29 October 2025

References

1. Jordan MI, Mitchell TM. Machine learning: trends, perspectives, and prospects. Science. 2015;349(6245):255–60.
2. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553):436–44.
3. Mitchell TM. Machine learning. New York: McGraw-Hill; 1997.
4. Bishop CM. Pattern recognition and machine learning. Cambridge: Springer; 2006.
5. Vapnik VN. The nature of statistical learning theory. New York City, NY: Springer; 1995. https://doi.org/10.1007/978-1-4612-2000-9.
6. Sutton RS, Barto AG. Reinforcement learning: an introduction. Cambridge: MIT Press; 2018.
7. Shalizi CR. Advanced data analysis from an elementary point of view. 2013. https://www.stat.cmu.edu/~cshalizi/ADAfaEPoV/. Accessed 30 Apr 2024.
8. AlQuraishi M. End-to-end differentiable learning of protein structure. Cell Syst. 2019;8(4):292–301.
9. Boulent J, Foucher S, Théau J, St-Charles PL. Convolutional neural networks for the automatic identification of plant diseases. Front Plant Sci. 2019;10:941.
10. Camacho DM, Collins KM, Powers RK, Costello JC, Collins JJ. Next-generation machine learning for biological networks. Cell. 2018;173(7):1581–92.
11. Karar ME, Alsunaydi F, Albusaymi S, Alotaibi S. A new mobile application of agricultural pests recognition using deep learning in cloud computing system. Alexandria Eng J. 2021;60(5):4423–32.
12. Libbrecht MW, Noble WS. Machine learning applications in genetics and genomics. Nat Rev Genet. 2015;16(6):321–32.
13. Olden JD, Lawler JJ, Poff NL. Machine learning methods without tears: a primer for ecologists. Q Rev Biol. 2008;83(2):171–93.
14. Senior AW, Evans R, Jumper J, Kirkpatrick J, Sifre L, Green T, et al. Improved protein structure prediction using potentials from deep learning. Nature. 2020;577(7792):706–10.
15. Swan AL, Mobasheri A, Allaway D, Liddell S, Bacardit J. Application of machine learning to proteomics data: classification and biomarker identification in postgenomics biology. OMICS. 2013;17(12):595–610.
16. Thessen AE. Adoption of machine learning techniques in ecology and earth science. One Ecosyst. 2016;1:e8621.
17. MacEachern SJ, Forkert ND. Machine learning for precision medicine. Genome. 2021. https://doi.org/10.1139/gen-2021-0042.
18. Quazi S. Artificial intelligence and machine learning in precision and genomic medicine. Med Oncol. 2022. https://doi.org/10.1007/s12032-022-01642-1.
19. Schmidt B, Hildebrandt A. Deep learning in next-generation sequencing. Drug Discov Today. 2021. https://doi.org/10.1016/j.drudis.2020.10.001.
20. Leung MK, Delong A, Alipanahi B, Frey BJ. Machine learning in genomic medicine: a review of computational problems and data sets. Proc IEEE. 2015. https://doi.org/10.1109/JPROC.2015.2394485.
21. Parekh VS, Jacobs MA. Integrated radiomic framework for breast cancer and tumor biology using advanced machine learning and multiparametric MRI. NPJ Breast Cancer. 2017. https://doi.org/10.1038/s41523-017-0049-3.
22. Cheng SH, Augustin C, Bethel A, Gill D, Anzaroot S, Brun J, et al. Using machine learning to advance synthesis and use of conservation and environmental evidence. Conserv Biol. 2018. https://doi.org/10.1111/cobi.13098.
23. Tuia D, Kellenberger B, Beery S, Costelloe BR, Zuffi S, Risse B, et al. Perspectives in machine learning for wildlife conservation. Nat Commun. 2022. https://doi.org/10.1038/s41467-022-29487-2.
24. Galal et al. Galal A, Talal M, Moustafa A. Applications of machine learning in metabolomics: Diseasemodeling and classification. Front Genet. 2022;13:1017340. https://doi.org/10.3389/fgene.2022.1017340.

25. Li S, Hua H, Chen S. Graph neural networks for single-cell omics data: a review of approaches and applications. Brief Bioinform. 2025. https://doi.org/10.1093/bib/bbaf109.
26. Elith J, Leathwick JR. Species distribution models: ecological explanation and prediction across space and time. Annu Rev Ecol Evol Syst. 2009;40:677–97.
27. Benos L, Tagarakis AC, Dolias G, Berruto R, Kateris D, Bochtis D. Machine learning in agriculture: a comprehensive updated review. Sensors. 2021;21(11):3758.
28. Tarca AL, Carey VJ, Chen XW, Romero R, Drăghici S. Machine learning and its applications to biology. PLoS Comput Biol. 2007. https://doi.org/10.1371/journal.pcbi.0030116.
29. Banerjee J, Taroni JN, Allaway RJ, Prasad DV, Guinney J, Greene C. Machine learning in rare disease. Nat Methods. 2023;20(6):803–14.
30. Butler KT, Davies DW, Cartwright H, Isayev O, Walsh A. Machine learning for molecular and materials science. Nature. 2018;559(7715):547–55.
31. Bzdok D, Krzywinski M, Altman N. Machine learning: a primer. Nat Methods. 2017;14(12):1119.
32. Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: data mining, inference, and prediction. New York City: Springer; 2009. https://doi.org/10.1007/978-0-387-84858-7.
33. Smith JR, Hendershot JN, Nova N, Daily GC. The biogeography of ecoregions: descriptive power across regions and taxa. J Biogeogr. 2020. https://doi.org/10.1111/jbi.13871.
34. Salvatier J, Wiecki TV, Fonnesbeck C. Probabilistic programming in Python using PyMC3. PeerJ Comput Sci. 2016. https://doi.org/10.7717/peerj-cs.55.
35. Tao Q, Barba-Montoya J, Huuki LA, Durnan MK, Kumar S. Relative efficiencies of simple and complex substitution models in estimating divergence times in phylogenomics. Mol Biol Evol. 2020. https://doi.org/10.1093/molbev/msaa049.
36. R Core Team. R: a language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing. 2024. https://www.R-project.org/. Accessed 30 Apr 2024.
37. Seabold S, Perktold J. statsmodels: Econometric and statistical modeling with Python. In: Proceedings of the 9th Python in Science Conference. 2010. Accessed 30 April 2024.
38. Cortes C, Vapnik V. Support-vector networks. Mach Learn. 1995;20(3):273–97. https://doi.org/10.1007/BF00994018.
39. Schölkopf B, Smola AJ. Learning with kernels: support vector machines, regularization, optimization, and beyond. Cambridge, MA: MIT Press; 2002.
40. Hsu CW, Chang CC, Lin CJ. A practical guide to support vector classification. 2003. https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf. Accessed 30 Apr 2024.
41. Noble WS. What is a support vector machine? Nat Biotechnol. 2006. https://doi.org/10.1038/nbt1206-1565.
42. Das D, Singh M, Mohanty SS, Chakravarty S. Leaf disease detection using support vector machine. In: 2020 International Conference on Communication and Signal Processing (ICCSP). IEEE. 2020:1036–40.
43. Yaşar Ş, Çolak C, Yologlu S. Artificial intelligence-based prediction of COVID-19 severity on the results of protein profiling. Comput Methods Programs Biomed. 2021. https://doi.org/10.1016/j.cmpb.2021.105996.
44. Young F, Rogers S, Robertson DL. Predicting host taxonomic information from viral genomes: a comparison of feature representations. PLoS Comput Biol. 2020;16(5):e1007894.
45. Meyer D, Dimitriadou E, Hornik K, Weingessel A, Leisch F. e1071: Misc functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. R package version 1.7–16. 2024. https://CRAN.R-project.org/package=e1071. Accessed 30 Apr 2024.
46. Kuhn M, Wickham H. Tidymodels: a collection of packages for modeling and machine learning using tidyverse principles. 2020. https://www.tidymodels.org. Accessed 30 Apr 2024.
47. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: machine learning in Python. J Mach Learn Res. 2011;12(85):2825–30.
48. Cock PJA, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, et al. Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics. 2009. https://doi.org/10.1093/bioinformatics/btp163.
49. Breiman L. Random forests. Mach Learn. 2001. https://doi.org/10.1023/A:1010933404324.
50. Epifanio I. Intervention in prediction measure: a new approach to assessing variable importance for random forests. BMC Bioinformatics. 2017. https://doi.org/10.1186/s12859-017-1650-8.
51. Brieuc MSO, Buehler DM, McCarthy EM. A practical introduction to random forest for genetic association studies in ecology and evolution. Mol Ecol Resour. 2018. https://doi.org/10.1111/1755-0998.12892.
52. Fabris A, Pereira TF, Ferreira CM. A new approach for interpretation of random forest models and its application to the biology of aging. Bioinformatics. 2018. https://doi.org/10.1093/bioinformatics/bty233.
53. Fraimout A, Debat V, Fellous S, Hufbauer RA, Foucaud J, Pudlo P, et al. Deciphering the routes of invasion of Drosophila suzukii by means of ABC random forest. Mol Biol Evol. 2017. https://doi.org/10.1093/molbev/msx050.
54. Chipman HA, George EI, McCulloch RE. BART: bayesian additive regression trees. Ann Appl Stat. 2010;4(1):266–98.
55. Liaw A, Wiener M. Classification and regression by randomForest. R News. 2002;2(3):18–22. https://CRAN.R-project.org/doc/Rnews/. Accessed 30 Apr 2024.
56. Nakhale A, Rathish Kumar BV. Comparison of activation functions in brain tumour segmentation using deep learning. In: Tripathi AK, Anand D, Nagar AK, editors. Proceedings of World Conference on Artificial Intelligence: Advances and Applications. WWCA 1997. Algorithms for Intelligent Systems. Singapore: Springer. 1997. https://doi.org/10.1007/978-981-99-5881-8_311-8_14.
57. Zhang D, Chen H, Zulfiqar H, Yuan S, Huang Q, Zhang Z, et al. iBLP: an XGBoost-based predictor for identifying bioluminescent proteins. Comput Math Methods Med. 2021. https://doi.org/10.1155/2021/6664362.
58. Chen T, Guestrin C. XGBoost: a scalable tree boosting system. Proc ACM SIGKDD Int Conf Knowl Discov Data Min. 2016. https://doi.org/10.1145/2939672.2939785.
59. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 1997. https://doi.org/10.1093/nar/25.17.3389.
60. Ridgeway G, Edwards D, Kriegler B, Schroedl S, Southworth H, Greenwell B, Boehmke B, Cunningham J. gbm: Generalized boosted regression models. R package version 2.2.2. 2024. https://CRAN.R-project.org/package=gbm. Accessed 30 Apr 2024.
61. Domingos P. A few useful things to know about machine learning. Communications of the ACM. 2012;55(10):78–87. https://doi.org/10.1145/2347736.2347755.
62. James G, Witten D, Hastie T, Tibshirani R, Taylor J. An introduction to statistical learning with applications in Python. New York City: Springer; 2023. https://doi.org/10.1007/978-1-0716-0202-1.
63. Kapoor S, Narayanan A. Leakage and the reproducibility crisis in machine-learning-based science. Patterns. 2023. https://doi.org/10.1016/j.patter.2023.100804.
64. Leban G, Zupan B, Vidmar G, Bratko I. Vizrank: data visualization guided by machine learning. Data Min Knowl Discov. 2006. 0.1007/s10618-005-0031-5.
65. Li H, Ma K, Fu Z, Wang S, Qu H. Interactive machine learning by visualization: a small data solution. In: 2018 IEEE Int Conf Big Data (Big Data). 2018. https://doi.org/10.1109/BigData.2018.8621952. Accessed 02 June 2025.
66. Thomas JJ, Cook KA. Illuminating the path: the research and development agenda for visual analytics. Richland, WA: Pacific Northwest National Lab (PNNL). 2005; Report No.: PNNL-SA-45230.
67. Vellido A, Martín-Guerrero JD, Lisboa PJ. Seeing is believing: the importance of visualization in real-world machine learning applications. In: Proc 19th Eur Symp Artif Neural Netw Comput Intell Mach Learn (ESANN). 2011:219–26. Accessed 02 June 2025.
68. Chen V, Yang M, Cui W, Kim JS, Talwalkar A, Ma J. Applying interpretable machine learning in computational biology—pitfalls, recommendations and opportunities for new developments. Nat Methods. 2024. https://doi.org/10.1038/s41592-024-02359-7.
69. Rudin C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nat Mach Intell. 2019. https://doi.org/10.1038/s42256-019-0048-x.

70. Symonds MRE, Blomberg SP. A primer on phylogenetic generalised least squares. In: Garamszegi LZ, ed. Modern phylogenetic comparative methods and their application in evolutionary biology. Berlin, Germany: Springer. 2014:105 - 130. https://doi.org/10.1007/978-3-662-43550-2_5.
71. Cohen J, Cohen P, West SG, Aiken LS. Applied multiple regression/correlationanalysis for the behavioral science. New York: Routledge; 2013. https://doi.org/10.4324/9780203774441.
72. Lumley T, Diehr P, Emerson S, Chen L. The importance of the normality assumption in largepublic health data sets. Annu Rev Public Health. 2002;23:151–69. https://doi.org/10.1146/annurev.publhealth.23.100901.140546.
73. Laubach ZM, Murray EJ, Hoke KL, Safran RJ, Perng W. A biologist's guide to model selection and causal inference. Proc R Soc B. 2021. https://doi.org/10.1098/rspb.2020.2815.
74. Huang BF, Boutros PC. The parameter sensitivity of random forests. BMC Bioinformatics. 2016. https://doi.org/10.1186/s12859-016-1228-x.
75. Konstantinov AV, Utkin LV. Interpretable machine learning with an ensemble of gradient boosting machines. Knowl-Based Syst. 2021. https://doi.org/10.1016/j.knosys.2021.106993.
76. Natekin A, Knoll A. Gradient boosting machines, a tutorial. Front Neurorobot. 2013. https://doi.org/10.3389/fneur.2013.00021.
77. Lundberg SM, Lee S-I. A unified approach to interpreting model predictions. Adv Neural Inf Process Syst. 2017. https://doi.org/10.48550/arXiv.1705.07874.
78. Lundberg SM, Lee SI. Consistent feature attribution for tree ensembles. arXiv. 2018. https://doi.org/10.48550/arXiv.1706.06060.
79. Zhang W, Li R, Zhao J, Wang J, Meng X, Li Q. Miss-gradient boosting regression tree: a novel approach to imputing water treatment data. Appl Intell. 2023;53:22917–37. https://doi.org/10.1007/s10489-023-04828-6.
80. Guo Y, Liu Y, Oerlemans A, Lao S, Wu S, Lew MS. Deep learning for visual understanding: a review. Neurocomputing. 2016. https://doi.org/10.1016/j.neucom.2015.09.116.
81. Meenalochini M, Amudha P. Cauliflower plant disease prediction using deep learning techniques. In: Tripathi AK, Anand D, Nagar AK, editors. Proceedings of World Conference on Artificial Intelligence: Advances and Applications. WWCA 1997. Algorithms for Intelligent Systems. Singapore: Springer. 1997. https://doi.org/10.1007/978-981-99-5881-8_14.
82. Reddy BSJN, Yadav S, Venkatakrishnan R, Oviya IR. Comparison of deep learning approaches for DNA-binding protein classification using CNN and hybrid models. In: Tripathi AK, Anand D, Nagar AK, editors. Proceedings of World Conference on Artificial Intelligence: Advances and Applications. WWCA 1997. Algorithms for Intelligent Systems. Singapore: Springer. 1997. https://doi.org/10.1007/978-981-99-5881-8_7.
83. Solieman H, Sali S. Classification of arrhythmias using a pre-trained deep learning model with binary images of segmented ECG. J Russ Univ. 1998;26(2):120–7.
84. Lo SCB, Chan HP, Lin JS, Li H, Freedman MT, Mun SK. Artificial convolution neural network for medical image pattern recognition. Neural Netw. 1995;8(7–8):1201–14.
85. Sajda P. Machine learning for detection and diagnosis of disease. Annu Rev Biomed Eng. 2006;8(1):537–65.
86. Goga AB, Kadri C, Naroua H. Deep learning for the recognition of benign and malignant breast lesions in medical images. ARPN J Eng Appl Sci. 2023. https://doi.org/10.59018/0923264.
87. Sarraf S, DeSouza DD, Anderson J, Tofighi G. DeepAD: Alzheimer's disease classification via deep convolutional neural networks using MRI and fMRI. bioRxiv. 2016; https://doi.org/10.1101/070441.
88. Zhang Q, Chi X, Zhao D. Early diagnosis of Parkinson's disease based on deep learning. Comput Syst Appl. 2021;27:1–9.
89. Zhou SK, Greenspan H, Davatzikos C, Duncan JS, van Ginneken B, Madabhushi A, et al. A review of deep learning in medical imaging: imaging traits, technology trends, case studies with progress highlights, and future promises. Proc IEEE. 2021;109(5):820–38.
90. Wallach I, Dzamba M, Heifets A. AtomNet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. arXiv preprint. 2015. https://doi.org/10.48550/arXiv.1510.02855.
91. Unterthiner T, Mayr A, Klambauer G, Steijaert M, Ceulemans H, Wegner J, Hochreiter S. Deep learning as an opportunity in virtual screening. In: Proceedings of the Deep Learning Workshop at NIPS. Cambridge, MA: MIT Press. 2014. Accessed 30 Apr 2024.
92. Alipanahi B, Delong A, Weirauch MT, Frey BJ. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. Nat Biotechnol. 2015;33(8):831–8.
93. Wheway G, Mitchison HM. Opportunities and challenges for molecular understanding of ciliopathies—the 100,000 genomes project. Front Genet. 2019;10:127.
94. Poplin R, Chang PC, Alexander D, Schwartz S, Colthurst T, Ku A, et al. A universal SNP and small-indel variant caller using deep neural networks. Nat Biotechnol. 2018;36(10):983–7.
95. Quang D, Chen Y, Xie X. DANN: a deep learning approach for annotating the pathogenicity of genetic variants. Bioinformatics. 2014;31(5):761–3.
96. AlQuraishi M. Alphafold at CASP13. Bioinformatics. 2019;35(22):4862–5.
97. Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, et al. Applying and improving AlphaFold at CASP14. Proteins. 2021a. https://doi.org/10.1038/s41586-021-03819-2.
98. Ching T, Himmelstein DS, Beaulieu-Jones BK, Kalinin AA, Do BT, Way GP, et al. Opportunities and obstacles for deep learning in biology and medicine. J R Soc Interface. 2018. https://doi.org/10.1098/rsif.2017.0387.
99. Wang C, Li M, Smola AJ. Language models with transformers arXiv preprint. 2019. https://doi.org/10.48550/arXiv.1904.09408.
100. Brandusescu A. Artificial intelligence policy and funding in Canada: public investments, private interests. SSRN. 2021. https://doi.org/10.2139/ssrn.4089932. Accessed 30 Apr 2024.
101. McNabb NK, Christensen EW, Rula EY, Coombs L, Dreyer K, Wald C, et al. Projected growth in FDA-approved artificial intelligence products given venture capital funding. J Am Coll Radiol. 2021;21(4):617–23.
102. Rahkovsky I, Toney A, Boyack KW, Klavans R, Murdick DA. AI research funding portfolios and extreme growth. Front Res Metr Anal. 2021. https://doi.org/10.3389/frma.2021.630124.
103. Bae G, Bae DI, Kang M, Hwang SM, Kim SS, Seo B, et al. 3nm GAA technology featuring multi-bridge-channel FET for low power and high performance applications. IEEE Int Electron Devices Meet. 2018. https://doi.org/10.1109/IEDM.2018.8614629.
104. Chu CH. Accelerator-enabled communication middleware for large-scale heterogeneous HPC systems with modern interconnects. PhD Dissertation. Ohio State Univ. 2020. http://rave.ohiolink.edu/etdc/view?acc_num=osu1595451131152. Accessed 30 Apr 2024.
105. Huang L, Yu W, Ma W, Zhong W, Feng Z, Wang H, et al. A survey on hallucination in large language models: principles, taxonomy, challenges, and open questions. ACM Trans Inf Syst. 2025;43(2):1–55.
106. Lee GG, Latif E, Shi L, Zhai X. Gemini Pro defeated by GPT-4V: evidence from education. arXiv preprint. 2023. https://doi.org/10.48550/arXiv.2401.08660.
107. Thirunavukarasu AJ, Ting DSJ, Elangovan K, Gutierrez L, Tan TF, Ting DSW. Large language models in medicine. Nat Med. 2023. https://doi.org/10.1038/s41591-023-02448-8.
108. Webb S. Deep learning for biology. Nature. 2018;554(7693):555–7.
109. Teubner T, Flath CM, Weinhardt C, van der Aalst W, Hinz O. Welcome to the era of ChatGPT et al. The prospects of large language models. Bus Inf Syst Eng. 2023. https://doi.org/10.1007/s12599-023-00795-x.
110. Littmann M, Selig K, Cohen-Lavi L, Frank Y, Hönigschmid P, Kataka E, et al. Validity of machine learning in biology and medicine increased through collaborations across fields of expertise. Nat Mach Intell. 2020;2:18–24.
111. Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, et al. Highly accurate protein structure prediction with AlphaFold. Nature. 2021. https://doi.org/10.1038/s41586-021-03819-2.
112. Watson JL, Juergens D, Bennett NR, Trippe BL, Yim J, Eisenach HE, et al. De novo design of protein structure and function with RFdiffusion. Nature. 2023;620(7976):1089–100.
113. Cui H, Wang C, Maan H, Pang K, Luo F, Duan N, et al. ScGPT: toward building a foundation model for single-cell multi-omics using generative AI. Nat Methods. 2024;21:1470–80.
114. Caton S, Haas C. Fairness in machine learning: a survey. ACM Comput Surv. 2024. https://doi.org/10.1145/3616865.

115. Jo ES, Gebru T. Lessons from archives: strategies for collecting sociocultural data in machine learning. Proc Conf Fairness Accountab Transpar. 2020. https://doi.org/10.1145/3351095.3372829.

Publisher's Note