International Journal of Data Science and Analytics (2026) 22:256
https://doi.org/10.1007/s41060-026-01246-y

REVIEW

Eleven quick tips to reduce overfitting in machine learning

Davide Chicco[1,2] · **Luca Oneto**[3]

Received: 8 June 2026 / Accepted: 23 July 2026

Abstract

Overfitting is the excessive adaptation of a machine learning model to its training data and remains a persistent challenge in biomedical informatics. In supervised learning, models may capture patterns "too well," failing to generalize to unseen data and yielding overly optimistic performance estimates. The problem is especially acute in biomedical settings, where datasets are high-dimensional, heterogeneous, and often of few samples. Numerous strategies have been proposed to mitigate overfitting in bioinformatics and health informatics. However, even established techniques can produce misleading results if misapplied, for example through data leakage, excessive hyperparameter tuning, inappropriate preprocessing, or inadequate validation. To address these pitfalls, we present eleven practical tips for reducing unintentional overfitting in supervised biomedical machine learning studies. The recommendations stress principled data splitting, domain-informed preprocessing, controlled model complexity, systematic tuning, comprehensive performance evaluation, and robustness analysis. Rather than offering an exhaustive treatment, we provide an accessible, practice-oriented guide to support more reliable and reproducible machine learning research. Although developed for biomedical informatics, these quick tips are broadly applicable across disciplines using supervised machine learning.

Keywords Machine learning · Overfitting · Supervised machine learning · Binary classification · Regression analysis · Recommendations · Guidelines

Abbreviations

AUPRC	Area under the precision–recall curve
AUROC	Area under the receiver characteristic curve
EHRs	Electronic health records
GEO	Gene Expression Omnibus
FIt-SNE	Fourier transform-accelerated interpolation-based t-SNE
GO	Gene Ontology
GRID	General Repository for Interaction Data Sets
i.i.d	Independent and identically distributed
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ML	Machine learning
MSE	Mean Squared Error
PCA	Principal component analysis
QC	Quality control
R^2	Coefficient of determination
RMSE	Root Mean Squared Error
SRA	Sequence Read Archive
SMAPE	Symmetric Mean Absolute Percentage Error
SNE	Stochastic Neighbor Embedding
TCGA	The Cancer Genome Atlas
TCIA	The Cancer Imaging Archive
UC Irvine ML Repo	University of California Irvine Machine Learning Repository
UMAP	Uniform Manifold Approximation and Projection
XAI	Explaniable artificial intelligence

✉ Davide Chicco
davide.chicco@unimib.it

[1] Università di Milano-Bicocca, Milan, Italy

[2] University of Toronto, Toronto, ON, Canada

[3] Università di Genova, Genoa, Italy

Introduction

Machine learning (ML) is now widely applied in the biomedical sciences for diagnosis, prognosis, biomarker discov-

Published online: 04 August 2026

ery [1], and treatment recommendation [2]. Despite advances in model capacity and empirical performance, biomedical machine learning remains highly vulnerable to *overfitting* [3–9]. Overfitting arises when a model adapts too closely to patterns in a finite training sample rather than to the underlying data-generating distribution, producing overly optimistic performance estimates that fail to generalize.

Overfitting is closely related to practices such as *over-validation* [10], data leakage [11], *data snooping* [12], *data dredging* [13], and *p-hacking* [14]. Although differing in mechanism and intent, these issues share a common outcome: biased inference that does not replicate independently.

Importantly, overfitting extends beyond predictive accuracy. Any quantity estimated from finite data (including correlations that induce shortcuts [15] or inferred causal relationships [16]) may be overfitted. Here, however, we focus on predictive performance, as accuracy-based metrics remain the primary basis for machine learning evaluation in practice.

A distinction can be drawn between *intentional* [14] and *unintentional* [17] overfitting. The former may result from questionable research practices; the latter more commonly reflects high model flexibility, high-dimensional feature spaces, small sample sizes, and implicit researcher degrees of freedom. This article addresses unintentional overfitting.

Overfitting is particularly concerning because it yields misleadingly optimistic estimates of future performance. By contrast, *underfitting* [18], while suboptimal, is often easier to detect and correct: The model fails to capture dominant patterns and performs poorly both during training and at deployment.

These risks are compounded by distribution shift: temporal changes, evolving clinical practice, instrumentation differences, diagnostic criteria updates, and population shifts [19, 20]. Although handling non-stationarity lies beyond the scope of this article, we adopt the classical *independent and identically distributed* (i.i.d.) assumption underlying most machine learning methodology. Even under this assumption, overfitting remains pervasive.

Extensive work has examined overfitting from statistical, computational, and applied perspectives [5, 21–24], alongside practical guidance on evaluation and reproducibility [25–27].

A few of these studies [22–24], in particular, focused on overfitting, providing general comments and considerations about this important aspect of machine learning.

The study by R. Roelofs and colleagues [22] presents an analysis of the overfitting phenomenon across 120 Kaggle competitions, investigating whether repeated use of a hold-out test set in supervised machine learning practice can lead to substantial overfitting. Surprisingly, the authors found almost no signs of adaptive overfitting across these different classification tasks, confirming the efficacy of hold-out validation in supervised computational intelligence.

In a tutorial study, J. Demšar and B. Zupan [23] provide a practical, hands-on introduction to understanding and recognizing overfitting in supervised machine learning, aimed especially at beginners and practitioners who need intuition beyond theoretical definitions. They present a series of exercises on how to recognize and address overfitting through the Orange open-source machine learning toolbox.

In their overview, X. Ying [24] provides a concise overview of overfitting in machine learning, explaining why it occurs, how it affects model performance, and the common approaches used to reduce it. The author describes its main causes (overly complex models with too many parameters, insufficient training data, noisy or irrelevant features, excessive training iterations, or inappropriate model selection), its main consequences, methods for detecting it, and potential solutions.

Although useful and interesting, these articles provide only a limited number of recommendations on how to address and mitigate overfitting, with relatively few actionable pieces of advice. In our article, instead, we provide a set of simple, targeted, and executable guidelines that can help readers understand overfitting, recognize it, and handle it efficiently, especially in the biomedical sciences.

The objective is not to eliminate overfitting (inevitable in finite-data settings) but to improve its recognition, quantification, and mitigation, thereby supporting more reliable predictive models and more credible scientific conclusions. In this study, we implicitly assume that the data consist of standard i.i.d. elements and that the training and test distributions are exchangeable, even though this assumption is often violated in biomedical settings involving temporal cohorts, multi-site studies, or longitudinal data.

We represent our quick tips in Fig. 1.

Tip 1: reserve a final untouched dataset for the very end

One of the fundamental principles of supervised machine learning is the strict independence between training and test sets: No data used for training should be used for testing, and vice versa. Despite being widely known, many published studies still suffer from data *leakage* [28] or *snooping* [29].

To mitigate overfitting, an untouched test set must be defined at the beginning of the project and accessed only once, at the very end. Model development, including all modeling decisions and hyperparameter tuning, should rely exclusively on training and validation data. The held-out test set must remain strictly isolated to provide an unbiased estimate of real-world performance. Properly enforcing this separation improves generalization and helps detect overfitting [17].

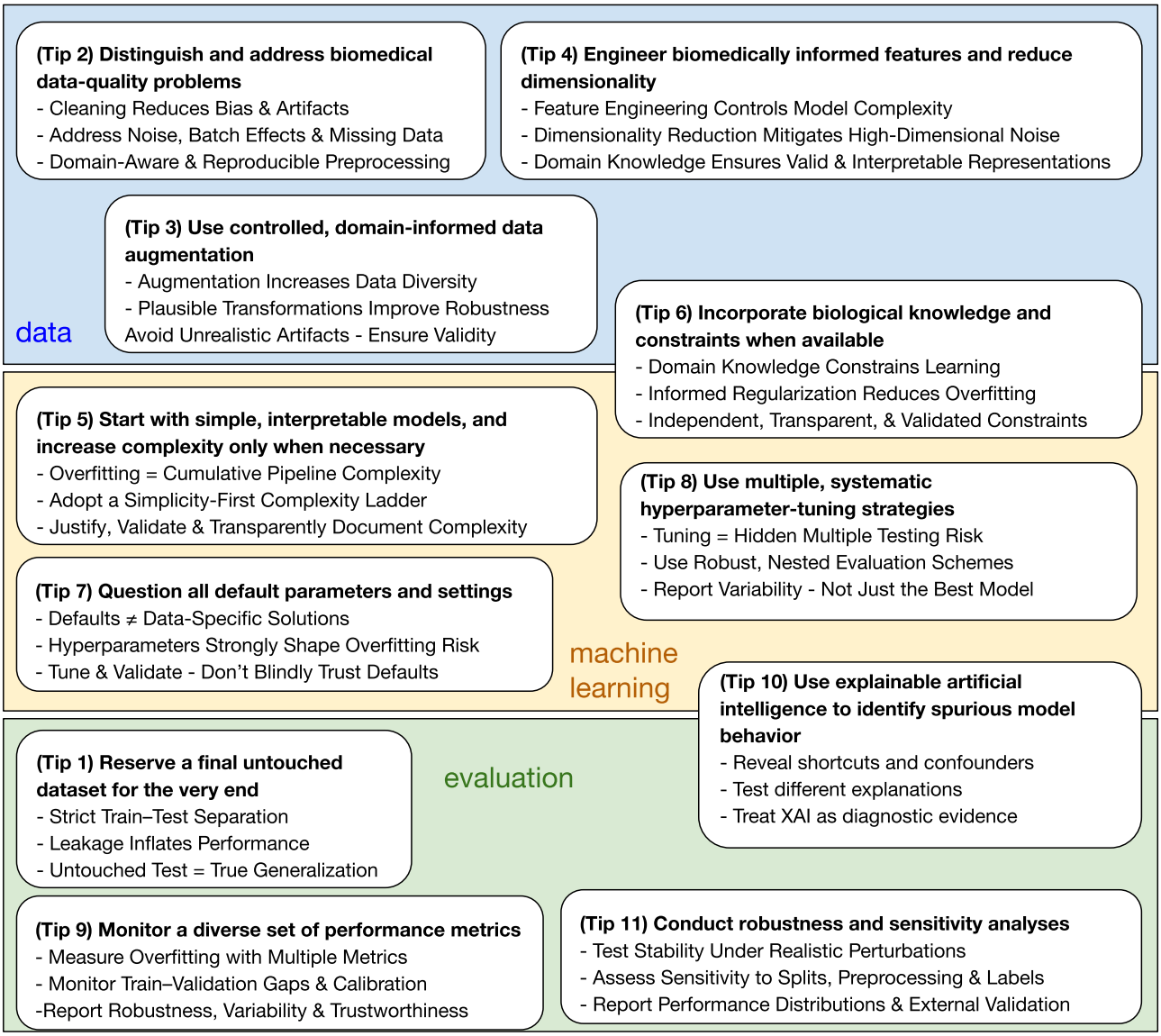

Fig. 1 Representation of our tips to reduce overfitting in supervised machine learning

Violations of this principle can produce misleading conclusions. For example, Michael Wainberg and colleagues [30] highlighted methodological issues in a study [31] evaluating several supervised learning algorithms on datasets from the University of California Irvine Machine Learning Repository. Similarly, Anton Bushuiev et al. [32] showed that many studies on protein–protein interactions suffer from leakage between training and test sets, often evaluating models on interactions already seen during training. They traced this problem to indirect transfer of metadata and sequence similarity, which contaminated performance estimates. Comparable concerns arise in bioinformatics applications involving cell types. The predictive performance of TargetFinder [33], a tool for enhancer–promoter interaction prediction, was questioned because its apparent accuracy may have reflected similarities across cell types rather than true predictive power [34, 35]. Since enhancer–promoter interactions can share patterns across cell types [36, 37], insufficient separation can again introduce leakage.

Although maintaining a fully untouched test set may seem unconventional in traditional statistical practice, where no formal training-test split is required, in machine learning it is essential. Among practical recommendations, this is arguably the most important.

Tip 2: distinguish and address biomedical data-quality problems

Data preprocessing, and data cleaning in particular, is a cornerstone of biomedical machine learning [38–42]. It converts raw, heterogeneous measurements into analysis-ready data through quality control, standardization of units and coding systems, consistency checks, and principled handling of erroneous or incomplete observations. However, data-quality problems should not be treated as a single category: Measurement noise, batch effects, outliers, and missing data arise through different mechanisms and therefore require different diagnostic and mitigation strategies.

Measurement noise consists of random or systematic deviations introduced by instruments, acquisition protocols, operators, environmental conditions, or biological sampling variability [43–46]. Noise can reduce signal-to-noise ratio and obscure reproducible associations, even when individual observations remain within plausible ranges. Appropriate responses may include instrument calibration, technical replication, signal filtering, repeated measurements, and automated quality-control procedures. The chosen method should reflect whether the noise is approximately random, heteroscedastic, correlated, or dependent on the measurement process.

Batch effects are structured, non-biological differences associated with processing dates, reagent lots, sequencing runs, scanners, laboratories, clinical sites, or other technical groups [47]. Unlike unstructured measurement noise, batch effects may systematically separate samples and can become confounded with outcomes or patient subgroups. They should be investigated using experimental metadata and exploratory analyses and addressed through balanced study design, normalization, harmonization, or metadata-informed adjustment. Batch correction must be applied carefully, because excessive correction can remove genuine biological or population-level variation.

Outliers and anomalies are observations that differ markedly from the rest of the dataset [48]. They may reflect data-entry errors, instrument failures, sample contamination, unusual acquisition conditions, or genuinely rare biological or clinical events. Outliers should therefore not be removed solely because they are statistically extreme. They should first be examined using robust statistics, anomaly-detection methods, source-data verification, and, where possible, expert review. Removal or modification is justified only when there is evidence that an observation is erroneous or incompatible with the intended analysis; otherwise, robust models or sensitivity analyses may be preferable.

Missing data arise when variables or observations are unavailable because of study design, censoring, patient dropout, failed assays, incomplete clinical documentation, or data-integration limitations [49]. Missingness is not equivalent to noise or an outlier and may itself carry clinical or operational information. Researchers should examine the extent, pattern, and likely mechanism of missingness and state the assumptions underlying their analysis. Possible strategies include complete-case analysis, explicit missingness indicators, single or multiple imputation, and models that accommodate incomplete observations [50]. No method is universally appropriate, particularly when missingness is related to the outcome or to unobserved patient characteristics.

These problems may interact. For example, a batch-specific instrument failure may produce both increased noise and missing values, while a measurement error may appear as an outlier. Nevertheless, their distinct origins should remain explicit, because combining them under a single cleaning procedure can conceal important assumptions and introduce bias.

Effective preprocessing must integrate domain expertise to preserve biological and clinical validity [51–53]. Plausibility constraints, such as physiological ranges, unit harmonization, and careful reconciliation of identifiers and ontologies, can prevent subtle inconsistencies from propagating into the model. In genomics and multi-site studies, the objective is to reduce technical variation without suppressing genuine biological signals or between-sample variability. Excessive filtering may improve apparent data "tidiness" while eliminating rare but meaningful subtypes, adverse events, or mechanistic deviations.

All data-dependent cleaning operations must be estimated using the training data only and then applied unchanged to validation and test data. For example, imputation values, outlier thresholds, normalization parameters, and batch-adjustment models should be learned within each training fold. Estimating these quantities from the complete dataset can introduce information leakage and produce optimistically biased performance estimates.

Finally, preprocessing should be rigorously documented to ensure transparency and reproducibility [54, 55]. Reports should separately describe the identification and treatment of measurement noise, batch effects, outliers, and missing data; justify any excluded or modified observations; state the assumptions underlying each procedure; and report the exact software versions and parameters used. When feasible, sharing preprocessing scripts, metadata schemas, exclusion logs, and quality-control summaries creates an audit trail for replication and critical evaluation.

Tip 3: use controlled, domain-informed data augmentation

Data augmentation [56] expands the effective training set by transforming existing samples or generating synthetic

ones. When designed using biological, clinical, or technical knowledge, augmentation can reduce overfitting, improve robustness, and enhance generalization by exposing models to plausible sources of variation without requiring additional data collection. Controlled, domain-informed augmentation is therefore particularly valuable in biomedical applications, where datasets are often small, costly to acquire, or insufficiently representative of the variability encountered in practice.

Effective augmentation should encode plausible invariances or variations in the underlying data-generating process. For images, commonly used operations include rotations, translations, scaling, cropping, flipping, and intensity perturbations; for tabular data, approaches may include carefully calibrated noise injection or synthetic over-sampling [56]. When these transformations preserve the target label and clinically relevant structures, they encourage models to learn deployment-relevant features rather than incidental characteristics of the training set.

Biomedical imaging provides several examples in which controlled augmentation can be beneficial [57]. Moderate geometric or intensity transformations may reproduce realistic variability associated with patient positioning, illumination, image reconstruction, scanner settings, or acquisition protocols. The appropriateness of a transformation nevertheless depends on the task: A rotation or reflection that is reasonable for one anatomical structure may be clinically implausible or may change the interpretation of another. Augmentation policies should therefore be selected with reference to anatomy, acquisition procedures, and the intended clinical use.

Domain-informed augmentation can also be useful in genomics and other molecular data. For example, perturbations may model realistic variation in sequencing depth, measurement noise, dropout, batch-related effects, or moderate expression fluctuations [58]. Sequence-based augmentation may likewise incorporate transformations that preserve known functional or structural properties. Such approaches can improve robustness when the assumed transformations are supported by biological knowledge and do not erase signals associated with the prediction target.

Generative modeling further extends augmentation by synthesizing samples that approximate the observed data distribution [59, 60]. These methods can generate novel instances or interpolations and may be particularly useful for rare conditions, under-represented subgroups, or settings in which data acquisition is expensive. Generative augmentation may thus broaden coverage of relevant biological and technical variability, provided that the generated samples are sufficiently realistic, diverse, and consistent with established clinical or biological constraints.

The benefits of augmentation are not automatic. Poorly chosen transformations can distort genuine biological signals, alter target semantics, or introduce implausible artifacts [51, 52]. They may consequently produce models that perform well on augmented data but fail to generalize or learn misleading associations. Augmentation parameters should therefore be treated as modeling choices that require empirical and domain-specific justification rather than as universally valid preprocessing defaults.

Particular care is required for data derived from electronic health records (EHRs). If an EHR dataset disproportionately represents a particular population, a conventional augmentation or synthetic-data procedure may reproduce or amplify that imbalance [61]. Synthetic data generated from biased source data may consequently perpetuate healthcare disparities in downstream clinical decision-making. However, carefully designed augmentation may also be used to improve representation of clinically meaningful and under-represented groups, provided that this objective is explicitly defined and that subgroup performance, calibration, and fairness are independently evaluated.

To ensure scientific validity, augmentation policies should be developed with relevant domain expertise and evaluated against a no-augmentation baseline. Useful controls include ablation studies, sensitivity analyses over transformation types and magnitudes, visual or biological plausibility checks, subgroup analyses, and validation on non-augmented and preferably external data. All transformations, parameter ranges, selection criteria, and validation procedures should be documented transparently to support reproducibility [54, 55].

Tip 4: engineer biomedically informed features and reduce dimensionality

Biomedical feature engineering and dimensionality reduction are important tools for mitigating overfitting. Feature engineering transforms raw variables into informative attributes tailored for supervised machine learning [62]. Typical operations include normalization and scaling, aggregation or combination of variables, categorical encoding, and feature selection. In biomedical contexts, these transformations must be guided by domain knowledge to ensure clinical and biological validity. For example, Ali et al. [63] applied feature engineering to reduce overfitting in machine learning models trained on diabetes data from electronic medical records.

When datasets contain extremely high-dimensional inputs, such as the ∼20,000 genes measured in single-cell RNA-seq studies [64], dimensionality reduction becomes essential for noise control and visualization. Methods such as PCA [65], t-SNE [66] (including FIt-SNE [67] and art-SNE [68]), UMAP [69], TriMap [70], PaCMAP [71], ForceAtlas [72], and PHATE [73] project high-dimensional data into lower-

dimensional spaces while attempting to preserve local and/or global structure. For instance, layerUMAP [74] used UMAP to compress deep learning representations of bioinformatics data, contributing to reduced overfitting.

There is no consensus on which method best preserves local versus global structure [75], and quantitative metrics for evaluating embedding quality have only recently been proposed [76, 77]. Importantly, low-dimensional visualizations are not equivalent to unsupervised clustering, although the two are sometimes conflated.

When applied appropriately and in conjunction with domain expertise, feature engineering and dimensionality reduction can reduce noise, control model complexity, and thereby help mitigate overfitting.

Tip 5: start with simple, interpretable models, and increase complexity only when necessary

Unintentional overfitting in biomedical machine learning rarely results from a single complex model; it more often reflects the *cumulative degrees of freedom* introduced throughout the pipeline - preprocessing, feature engineering, model selection, hyperparameter tuning, early stopping, ensembling, and thresholding [5]. A practical safeguard is a *complexity ladder*: start with the simplest scientifically defensible baseline and increase complexity only when it provides stable, reproducible improvements under an unchanged evaluation protocol [23, 78].

Model complexity is multi-dimensional [79, 80]: intrinsic flexibility (capacity), procedural choices (which may act as implicit multiple testing [14]), data requirements, and reproducibility burden. Interpretable baselines (for example, regularized linear, logistic, or Cox models) anchor expectations and can expose leakage when "simple" models perform implausibly well [11]. Additional complexity should be justified a priori and supported by robust (not marginal) gains. Predefine model families and search spaces, avoid repeated reuse of validation data, use nested cross-validation [81] or a final untouched test set, and report variability across folds [82].

Pre-trained or foundation models may reduce labeled-data needs, but they also introduce the risk of hidden contamination if related samples, benchmarks, institutions, or proxy variables were represented during pre-training [11]. Before using such models in a clinical setting, researchers should conduct a contamination audit. At minimum, this audit should ask:

(i) whether the pre-training corpus may overlap with the target task, cohort, or benchmark;
(ii) whether exact duplicates, near-duplicates, repeated patients, or repeated acquisitions could exist between downstream and pre-training-related data;
(iii) whether performance persists on temporally separated and geographically external cohorts;
(iv) whether results remain stable after removing metadata or potentially identifying shortcut variables;
(v) whether gains remain when compared with simple interpretable baselines under patient-level, site-level, or time-based splits.

Practical diagnostic tests include duplicate screening, external and temporal validation, ablation of metadata and proxy variables, and sensitivity analyses across alternative splitting strategies. Sharp performance declines under these checks should be treated as possible evidence of contamination and reported transparently.

Of course, we are not suggesting avoiding complex machine learning models *tout court*: Sometimes these methods are the only computational algorithms capable of solving a specific biomedical task. Examples include radiological tumor detection and segmentation, medical image segmentation, and digital pathology and histopathological image analysis, where convolutional neural networks (CNNs), ResNet, DenseNet, and other complex deep learning methods are among the few approaches capable of achieving good results. Model complexity should be justified by evidence rather than adopted by habit.

Transparent documentation remains essential for reproducibility and trust [54, 55].

Tip 6: incorporate biological knowledge and constraints when available

Overfitting in biomedical machine learning often arises when highly flexible models are trained on limited data with weak or noisy biological guidance. Reliable domain knowledge should therefore be treated as *information* that constrains learning (via priors, penalties, architectures, or mechanistic bounds) thereby reducing effective degrees of freedom, discouraging shortcut solutions, and improving generalization [15, 51, 52].

Such constraints act as informed regularization, shrinking the hypothesis space and variance [79, 80]. Because incorrect or context-mismatched knowledge can introduce bias, *soft* constraints are generally preferable; *hard* constraints should be reserved for physical laws or definitional truths (for example, non-negativity or conservation).

Knowledge can be incorporated during preprocessing, training, or postprocessing, with priority given to constraints embedded during training [51, 52]. Examples include biologically informed feature construction (for example, pathway

scores), structure-aware regularization aligned with known groups or networks, architectural inductive biases restricting interactions to curated graphs, mechanistic penalties enforcing physiological consistency, and output-level coherence constraints for clinically compatible predictions [83, 84].

Constraints must not be derived using outcome labels or tuned through repeated validation-set interaction [11]. They should rely on independent knowledge sources, with any data-driven structure learned strictly within training folds and evaluated on an untouched test set.

Constrained models should be compared to unconstrained baselines, reporting variability across folds and runs. Negative controls (for example, randomized pathways) and effects on calibration, subgroup robustness, and external validity should be assessed [27]. Transparent documentation of constraint sources, versions, implementation (hard versus soft), and sensitivity analyses is essential [54, 55]. When scientifically adequate, simpler constraints should be preferred.

Hybrid mechanistic and data-driven modeling approaches can also be valid alternatives to reduce overfitting [85], which can be combined with the tips presented here [86].

Tip 7: question all default parameters and settings

The widespread availability of machine learning libraries in Python, R, and Julia has encouraged a "plug-and-play" habit: load the data, run the default function, and observe the output. However, ignoring software settings and hyperparameters can substantially increase the risk of overfitting [87].

Default hyperparameters and preprocessing choices are designed for broad applicability, not for the specific characteristics of a given biomedical dataset. Biomedical data often differ markedly from benchmark datasets and may require stronger regularization or tighter complexity control. Defaults can imply weak regularization or excessive model flexibility, leading to convergence toward overfitted solutions.

For instance, the default maximum tree depth in Random Forests [88] may be unnecessarily large, increasing overfitting risk [89, 90]. In practice, non-experts often underestimate how strongly hyperparameters influence model behavior and study conclusions.

Other concrete examples of default parameters and hyperparameters that can strongly influence machine learning performance and overfitting risk include: regularization strength as C hyperparameter in logistic regression and linear regression and as weight decay in artificial neural networks; learning rate in gradient boosting and artificial neural networks; number of estimators in Random Forests and boosting algorithms; the C hyperparameter as trade-off between classification errors and margin width, kernel, and gamma for Support Vector Machines.

Defaults should not be blindly discarded, but neither should they be accepted uncritically. They must be examined, understood, and empirically validated through principled tuning and proper evaluation, selecting values appropriate for the data and scientific objective.

For model diagnostic, we suggest to test a set of values for each hyperparameter, and then check the corresponding learning curves.

Tip 8: use multiple, systematic hyperparameter-tuning strategies

Hyperparameter tuning is a major source of unintentional overfitting: each additional "knob" effectively introduces another look at the data. Repeated querying of a validation set turns model selection into a multiple-testing problem, inflating reported performance even without explicit leakage [5, 10, 14].

Evaluation schemes should therefore be chosen deliberately. Single hold-out validation is simple but high-variance and split-sensitive [82]. Repeated hold-out and k-fold cross-validation improve stability, while repeated k-fold further reduces variance. Leave-one-out maximizes training size but can be unstable in small, noisy biomedical datasets [91]. Bootstrap methods estimate optimism-corrected performance and uncertainty, provided the true sampling unit (for example, patient rather than visit) is respected [82, 92]. Comparing multiple resampling schemes on the same pipeline can reveal instability when results diverge.

No single optimization strategy suffices. Grid search suits small spaces [82], random search is more effective in high-dimensional settings [93], and Bayesian or other model-based approaches improve sample efficiency [94]. Evolutionary, gradient-based, reinforcement learning, or neural architecture search methods may be appropriate for complex landscapes or architectural tuning [95–98]. Selected configurations should perform consistently across resampling schemes, not only under a single evaluation setup. Adaptive data analysis strategies—such as capped tuning budgets or rotating validation splits—can further reduce overfitting to validation feedback [78, 99].

Finally, evaluation must remain properly nested: All preprocessing and tuning steps should occur strictly within training folds (nested cross-validation) or precede a single evaluation on a final untouched test set [11, 27, 82]. All explored configurations and performance variability (not just the best run) should be documented and reported [54, 55].

Table 1 Metrics to monitor the gap between training set and test set

supervised scientific problem	best metric	other useful metrics
binary classification	MCC	sensitivity, specificity, precision, NPV
regression analysis	R^2	SMAPE

These metrics must be monitored both on training set and test set: If the performance on the test set plateaus or worsens while training performance continues to improve, it signals overfitting. MCC: Matthews correlation coefficient [100, 101]. NPV: negative predictive value. R^2: coefficient of determination [102]. SMAPE: symmetric mean absolute percentage error

Tip 9: monitor a diverse set of performance metrics

Everything that needs to be improved must first be measured: Overfitting must be quantified to be mitigated. Relying on a single metric is insufficient, as each score reflects only one objective. Using multiple complementary metrics provides a more reliable and multi-dimensional assessment.

For classification, track training and validation loss, calibration, and confusion-matrix-derived measures such as Matthews correlation coefficient (MCC), sensitivity, specificity, precision, and negative predictive value. For regression, monitor R^2 and SMAPE (Table 1). Assess cohort-specific performance, robustness to class imbalance, and variability across folds. Avoid relying solely on the area under receiver characteristic curve (AUROC), the area under the precision–recall curve (AUPRC), F_1 score, accuracy, MAE, MAPE, MSE, and RMSE, as they can be misleading in imbalanced biomedical settings: These scores should be used with caution.

A widening gap between training and validation metrics is the clearest signal of overfitting. Early stopping (halting training when validation performance degrades or plateaus) can serve as a practical safeguard. While some metrics are useful during training, they may be inappropriate for final test evaluation; additional *ad hoc* indices have been proposed to quantify overfitting directly. Metrics of model trustworthiness should also be considered.

Everything that needs to be improved must first be measured: Overfitting must be quantified to be mitigated. Relying on a single metric is insufficient, as each score reflects only one objective [27]. Using multiple complementary metrics provides a more reliable and multi-dimensional assessment.

Recent work on *safe machine learning* broadens this principle from the monitoring of predictive performance alone to the quantitative assessment of multiple dimensions of model risk [103, 104]. In this literature, trustworthiness is commonly organized around sustainability or robustness, accuracy, fairness, and explainability. The Rank Graduation Box, for example, proposes a family of model-agnostic metrics derived from a common Lorenz- and concordance-curve framework, allowing these dimensions to be expressed on comparable scales [105]. The more recent integrated SAFE-AI framework evaluates model outputs under data removal, data poisoning, and feature removal to quantify accuracy, security or robustness, and explainability, respectively, through Rank Graduation Accuracy (RGA), Rank Graduation Robustness (RGR), and Rank Graduation Explainability (RGE) measures [106]. Related work has proposed Wasserstein-distance formulations for comparing models across the same broad SAFE dimensions, showing that multi-dimensional and perturbation-based evaluation need not depend on a single metric family [107].

This safe-machine-learning perspective is complementary to our focus on overfitting. Conventional predictive metrics quantify the training-validation generalization gap, whereas SAFE metrics evaluate whether model behavior remains acceptable under predefined perturbations and across several dimensions of trustworthiness. In biomedical studies, both levels should be considered: Task-specific metrics are needed to evaluate clinical predictive performance, while perturbation-based metrics can reveal instability, security vulnerabilities, or excessive dependence on individual features.

For classification, track training and validation loss, calibration, and confusion-matrix-derived measures such as Matthews correlation coefficient (MCC), sensitivity, specificity, precision, and negative predictive value [100, 101]. For regression, monitor R^2 and SMAPE [102] (Table 1). Assess cohort-specific performance, robustness to class imbalance, and variability across folds, random seeds, clinical sites, and relevant patient subgroups.

The RGA provides a response-agnostic, rank-based accuracy measure that can be applied to binary, ordinal, and continuous outcomes. However, for binary outcomes, the RGA coincides with the AUROC [108]. Consequently, although the RGA facilitates comparison within an integrated SAFE framework, it does not remove the limitations of relying exclusively on AUROC in imbalanced biomedical classification. Avoid relying solely on AUROC, the area under the precision–recall curve (AUPRC), F_1 score, accuracy, MAE, MAPE, MSE, or RMSE, as these measures can be misleading when used in isolation [100, 102, 109]. They should instead be interpreted alongside calibration, confusion-matrix-derived measures, subgroup results, uncer-

tainty estimates, and, when appropriate, clinically defined costs or harms.

A widening gap between training and validation metrics is one of the clearest signals of overfitting. Early stopping, which halts training when validation performance degrades or plateaus, can serve as a practical safeguard [110]. While some metrics are useful during training, they may be inappropriate for final test evaluation; additional *ad hoc* indices have therefore been proposed to quantify overfitting directly [22, 111, 112].

SAFE-style composite scores may be useful for model comparison, compliance monitoring, and checking whether pre-specified guardrails are crossed [106]. Nevertheless, the individual accuracy, robustness, fairness, and explainability components, together with their perturbation-response curves, should also be reported. A single aggregate score may otherwise conceal an unacceptable weakness in one safety-relevant dimension. Metrics of model trustworthiness should therefore complement, rather than replace, detailed task-specific evaluation [86].

Tip 10: use explainable artificial intelligence to identify spurious model behavior

Explainable artificial intelligence (XAI) methods can complement conventional performance evaluation by showing which variables, image regions, or patterns influence a model's predictions. In biomedical machine learning, these methods can help determine whether a model relies on clinically or biologically meaningful signals rather than on artifacts, confounders, or dataset-specific correlations.

XAI is particularly relevant to overfitting because a model may achieve high predictive performance while exploiting unintended shortcuts. For example, imaging models may rely on annotations, image borders, scanner characteristics, or hospital-specific acquisition patterns rather than on pathological structures. Similarly, models trained on electronic health records or genomic data may depend on missingness patterns, administrative variables, batch effects, or technical features instead of patient physiology or biological mechanisms. Explanation methods have been used to reveal such "Clever Hans" behavior and shortcut learning.

Explanations can therefore guide targeted diagnostic analyses. Researchers may compare explanations across data splits, random seeds, patient subgroups, clinical sites, and external datasets. Suspected shortcuts should be tested through feature ablation, image occlusion, counterfactual perturbation, negative controls, or removal of potentially confounding variables. Large changes in predictions after modifying a clinically irrelevant feature provide evidence that the model may have learned a spurious association.

However, XAI methods also have important limitations. Different methods may produce inconsistent explanations, and visually plausible saliency maps may not faithfully represent the fitted model. Moreover, feature importance does not establish causality, clinical validity, or fairness. Explanations should therefore be interpreted with domain expertise and treated as diagnostic evidence rather than as proof that a model is trustworthy.

Explainable artificial intelligence (XAI) methods can complement conventional performance evaluation by showing which variables, image regions, or patterns influence a model's predictions [113]. In biomedical machine learning, these methods can help determine whether a model relies on clinically or biologically meaningful signals rather than on artifacts, confounders, or dataset-specific correlations [114].

XAI is particularly relevant to overfitting because a model may achieve high predictive performance while exploiting unintended shortcuts. For example, imaging models may rely on annotations, image borders, scanner characteristics, or hospital-specific acquisition patterns rather than on pathological structures. Similarly, models trained on electronic health records or genomic data may depend on missingness patterns, administrative variables, batch effects, or technical features instead of patient physiology or biological mechanisms. Explanation methods have been used to reveal such "Clever Hans" behavior and shortcut learning [115, 116].

The safe-machine-learning literature gives XAI an additional role: explainability is treated as a measurable model-risk dimension rather than only as a visualization or qualitative interpretation. For example, the Shapley-Lorenz approach combines Shapley attribution with a normalized Lorenz-based variable-importance criterion [117]. The Rank Graduation Box subsequently places explainability on a scale that is consistent with its accuracy, fairness, and robustness measures [105]. Giudici and Kolesnikov further evaluate explainability through controlled feature-removal perturbations and integrate the resulting RGE measure with RGA and RGR in a unified SAFE-AI assessment [106].

These approaches complement, rather than replace, the diagnostic use of conventional XAI advocated here. Attribution methods can identify the variables or regions that appear to drive individual predictions, whereas SAFE-style perturbation analyses quantify how strongly model outputs change as features are removed and whether a predefined explainability or robustness guardrail is crossed. Using the two approaches together can therefore provide both a substantive explanation of model behavior and a quantitative stress test of its stability.

Explanations can guide targeted diagnostic analyses. Researchers may compare explanations across data splits, random seeds, patient subgroups, clinical sites, model families, and external datasets. The stability of both global feature rankings and local patient-level explanations should

be assessed. Suspected shortcuts should be tested through feature ablation, image occlusion, counterfactual perturbation, negative controls, or removal of potentially confounding variables. Large changes in predictions after modifying a clinically irrelevant feature provide evidence that the model may have learned a spurious association.

Feature-removal analyses must nevertheless be designed carefully. In biomedical datasets, predictors are often strongly correlated or constrained by biological, anatomical, or temporal relationships. Removing one feature independently may generate implausible observations, redistribute importance among correlated variables, or exaggerate the apparent dependence of the model on that feature. Whenever possible, perturbations should therefore be conditional, group-wise, or constrained by domain knowledge, and their biological or clinical plausibility should be verified.

XAI and SAFE explainability metrics also have important limitations. Different methods may produce inconsistent explanations, and visually plausible saliency maps may not faithfully represent the fitted model [118, 119]. Likewise, a favorable RGE value or integrated SAFE score does not establish that an explanation is causally valid, clinically meaningful, fair, or understandable to its intended users. Feature importance does not establish causality, and explanation stability does not by itself demonstrate external validity.

Explanations should therefore be interpreted with domain expertise and treated as diagnostic evidence rather than as proof that a model is trustworthy. For safety-relevant biomedical applications, explanation results should be reported together with predictive performance, calibration, subgroup analyses, robustness tests, uncertainty estimates, and external validation.

Tip 11: conduct robustness and sensitivity analyses and use external validation data

Robustness and sensitivity analyses extend standard validation by testing whether conclusions remain stable under *reasonable perturbations* of data, labels, and modeling choices. This is especially important in biomedical settings, where small, noisy, and heterogeneous datasets make single "best" estimates potentially misleading and prone to hidden overfitting [5, 10].

At minimum, robustness should assess variability across key uncertainty sources. *Alternative data splits* involve repeating partitions with different random seeds and structured holdouts (for example, patient-, site-, or time-based splits) to probe cohort sensitivity [120]. *Preprocessing choices* (normalization, batch correction, imputation, outlier handling) should be varied, ensuring all steps are fit strictly within training folds to avoid leakage [11]. *Input perturbations* (such as noise injection, simulated missingness, or calibration shifts) should cause gradual rather than catastrophic degradation [121, 122]. *Label definitions* and endpoints, particularly in EHR studies, should be tested for sensitivity to coding or time-window choices [123]. Finally, *modeling decisions* should be stress-tested through ablations, threshold variation, alternative class weighting, and reasonable hyperparameter ranges [124].

Robust models exhibit consistent rankings, bounded performance changes, stable calibration, and limited decision instability across perturbations. Results should be reported as distributions rather than single values, with subgroup-specific analyses when feasible [27]. Whenever possible, include at least one *external* dataset, as external validation often reveals artifacts undetected by internal resampling [19].

Using external datasets for validation is actually one of the most effective ways to identify overfitting and to test the generalization skills of a supervised machine learning model [29]. Public datasets can be be easily found on internet repositories: Gene Expression Omnibus (GEO), ArrayExpress, Sequence Read Archive (SRA), and the Cancer Genome Atlas (TCGA) for bioinformatics data, the Cancer Imaging Archive (TCIA) for medical images, PhysioNet for physiological signals, and Figshare, Zenodo, Kaggle, Hugging Face, and University of California Irvine Machine Learning Repository (UC Irvine ML Repo) for any data type. Public datasets can also be retrieved through Google Dataset Search, or found on specific data initiatives and repositories [125].

Obtaining good results on an external validation cohort, in fact, can prove the model works well not only on the original dataset, suggesting it could perform efficiently on other datasets too.

This approach helps identify models that have learned dataset-specific patterns, biases, or noise rather than generalizable relationships. In biomedical applications, where data distributions can vary substantially across hospitals and patient populations, external validation is particularly important to ensure model robustness, reliability, and potential clinical applicability.

In a computational biology example, R. Kustra and A. Zagdanski [126] used a data fusion strategy to perform clustering on microarray gene expression data and subsequently assigned the discovered clusters to Gene Ontology annotations (GO) [127, 128]. To validate their findings, instead of using another microarray dataset, they leveraged an external database of protein–protein interactions: the General Repository for Interaction Data Sets (GRID) [129]. This approach allowed the authors to confirm their results in an independent database, thereby strengthening the robustness of their scientific claims.

Robustness analyses should be pre-specified and treated as diagnostic (not optimization) procedures, with all outcomes

transparently reported to avoid overfitting to the robustness suite itself [78, 99].

Discussion

Unintentional overfitting in biomedical machine learning rarely arises from a single mistake; more often, it reflects the accumulation of small but consequential decisions across the full analytical pipeline. Common pitfalls include inadequate separation between development and evaluation data, preprocessing steps fit on the full dataset rather than within training folds, augmentation or feature manipulation without biomedical justification, unnecessary increases in pipeline complexity, uncritical reliance on software defaults, excessive hyperparameter tuning, selective emphasis on favorable metrics, and insufficient robustness analysis.

From a practical perspective, each stage of the workflow should be treated as a potential source of overfitting. Data splits must respect the true sampling structure and the final test set should remain untouched. Preprocessing, feature engineering, and dimensionality reduction must be estimated strictly within training data. Augmentation and biologically informed constraints should be used only when scientifically defensible. Added model or pipeline complexity should be justified by stable gains over simple baselines. Default settings should be checked critically, and hyperparameter tuning should remain limited, systematic, and nested within resampling. Evaluation should rely on multiple complementary metrics, and results should be supported by sensitivity analyses, subgroup analyses, and whenever possible external validation.

Overall, these pitfalls show that overfitting should be regarded as a property of the whole study design rather than of the final model alone. Transparent reporting, methodological restraint, and robustness-oriented evaluation are therefore essential for credible biomedical machine learning research.

Since overfitting is specific to each individual dataset and the overall algorithm setup, no dedicated software package exists to reduce it to date. However, several tutorials and online guides are available for Python [130] and for R [131–135], and they can be useful for beginners and novice computational researchers.

Conclusion

As Pedro Domingos noted, overfitting is the "bugbear" of supervised machine learning [136]. Despite its central importance, it remains frequently underestimated—particularly in biomedical research, where high dimensionality, small sample sizes, and heterogeneous data amplify its impact.

Many methodological tools can mitigate overfitting, but they are effective only when applied with clear awareness of the risks they address. In this article, we presented eleven practical recommendations to help researchers recognize, quantify, and reduce unintentional overfitting in supervised biomedical machine learning studies. These quick tips emphasize principled data splitting, biomedically informed preprocessing, controlled model complexity, careful hyperparameter tuning, comprehensive performance evaluation, and structured robustness analyses (Fig. 1).

These recommendations aim to mitigate (not eliminate) overfitting. No universal solution exists, especially in finite-data settings where some degree of overfitting is unavoidable. Progress therefore depends on methodological rigor, transparency, and critical self-evaluation throughout the analytical pipeline.

Although developed primarily for biomedical informatics and less experienced practitioners, these guidelines are equally relevant for experts and broadly applicable across disciplines using supervised machine learning. Wider awareness and consistent adoption can foster more reliable models, more reproducible results, and more credible scientific conclusions.

Acknowledgements None.

Author Contributions D.C. conceived the study, L.O. wrote most of the tips, both D.C. and L.O. supervised the study and contributed to the writing and to the reviewing.

Funding Open access funding provided by Università degli Studi di Milano - Bicocca within the CRUI-CARE Agreement. The work of D.C. is partially funded by the Italian Ministero Italiano delle Imprese e del Made in Italy under the Digital Intervention in Psychiatric and Psychologist Services (DIPPS) programme and is partially supported by Ministero dell'Università e della Ricerca of Italy under the "Dipartimenti di Eccellenza 2023-2027" ReGAInS grant assigned to Dipartimento di Informatica Sistemistica e Comunicazione at Università di Milano-Bicocca.
The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Data Availability No datasets were generated or analyzed during the current study.

Declarations

Ethical Approval Not applicable.

Consent to participate Not applicable.

References

1. Alum, E.U.: AI-driven biomarker discovery: enhancing precision in cancer diagnosis and prognosis. Discover Oncology **16**(1), 1–12 (2025). https://doi.org/10.1007/s12672-025-02068-5
2. Duwe, G., Mercier, D., Wiesmann, C., Kauth, V., Moench, K., Junker, M., Neumann, C.C.M., Haferkamp, A., Dengel, A., Höfner, T.: Challenges and perspectives in use of artificial intelligence to support treatment recommendations in clinical oncology. Cancer Med. **13**(12), e7398 (2024). https://doi.org/10.1002/cam4.7398
3. Aliferis, C., Simon, G.: Overfitting, underfitting and general model overconfidence and under-performance pitfalls and best practices in machine learning and AI. Artificial Intelligence and Machine Learning in Health Care and Medical Sciences (2024). https://doi.org/10.1007/978-3-031-47024-0_18
4. Gygi, J.P., Kleinstein, S.H., Guan, L.: Predictive overfitting in immunological applications: pitfalls and solutions. Human Vaccines & Immunotherapeutics **19**(2), 2251830 (2023). https://doi.org/10.1080/21645515.2023.2251830
5. Montesinos López, O.A., Montesinos López, A., Crossa, J.: Overfitting, model tuning, and evaluation of prediction performance. In Multivariate Statistical Machine Learning Methods for Genomic Prediction (2022). https://doi.org/10.1007/978-3-030-89010-0_6
6. Xu, C., Coen-Pirani, P., Jiang, X.: Empirical study of overfitting in deep learning for predicting breast cancer metastasis. Cancers **15**(7), 1969 (2023). https://doi.org/10.3390/cancers15071969
7. Ng, S., Masarone, S., Watson, D., Barnes, M.R.: The benefits and pitfalls of machine learning for biomarker discovery. Cell Tissue Res. **394**(1), 17–31 (2023). https://doi.org/10.1007/s00441-023-03800-9
8. Deo, R.C., Nallamothu, B.K.: Learning about machine learning: the promise and pitfalls of big data and the electronic health record. Circ. Cardiovasc. Qual. Outcomes **9**(6), 618–620 (2016). https://doi.org/10.1161/CIRCOUTCOMES.116.003090
9. Cai, Y.-Q., Gong, D.-X., Tang, L.-Y., Cai, Y., Li, H.-J., Jing, T.-C., Gong, M., Hu, W., Zhang, Z.-W., Zhang, X., Zhang, G.-W.: Pitfalls in developing machine learning models for predicting cardiovascular diseases: challenge and solutions. J. Med. Internet Res. **26**, e47645 (2024). https://doi.org/10.2196/47645
10. Cawley, G.C., Talbot, N.L.C.: On over-fitting in model selection and subsequent selection bias in performance evaluation. J. Mach. Learn. Res. **11**, 2079–2107 (2010). (https://www.jmlr.org/papers/v11/cawley10a.html)
11. 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
12. White, H.: A reality check for data snooping. Econometrica **68**(5), 1097–1126 (2000). https://doi.org/10.2307/2999432
13. Erasmus, A., Holman, B., Ioannidis, J.P.A.: Data-dredging bias. BMJ. Evid. Based Med. **27**(4), 209–211 (2022). https://doi.org/10.1136/bmjebm-2021-111800
14. Head, M.L., Holman, L., Lanfear, R., Kahn, A.T., Jennions, M.D.: The extent and consequences of p-hacking in science. PLoS Biol. **13**(3), e1002106 (2015). https://doi.org/10.1371/journal.pbio.1002106
15. Geirhos, R., Jacobsen, J., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., Wichmann, F.A.: Shortcut learning in deep neural networks. Nature Machine Intelligence **2**(11), 665–673 (2020). https://doi.org/10.1038/s42256-020-00257-z
16. Janzing, D.: Causal regularization. Neural Information Processing Systems, 2019. PDF URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/2172fde49301047270b2897085e4319d-Paper.pdf
17. Hosseini, M., Powell, M., Collins, J., Callahan-Flintoft, C., Jones, W., Bowman, H., Wyble, B.: I tried a bunch of things: the dangers of unexpected overfitting in classification of brain data. Neuroscience & Biobehavioral Reviews **119**, 456–467 (2020). https://doi.org/10.1016/j.neubiorev.2020.09.013
18. Cunningham, P., Delany, S.J.: Underestimation bias and underfitting in machine learning. In: International Workshop on the Foundations of Trustworthy AI Integrating Learning, Optimization and Reasoning, (2020). https://doi.org/10.1007/978-3-030-45691-7_4
19. Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., Lee, T., David, E., Stavness, I., Guo, W., Earnshaw, B., Haque, I., Beery, S. M., Leskovec, J., Kundaje, A., Pierson, E., Levine, S., Finn, C., Liang, P.: Wilds: a benchmark of in-the-wild distribution shifts. In International Conference on Machine Learning, (2021).https://doi.org/10.48550/arXiv.2012.07421
20. Bayram, F., Ahmed, B.S., Kassler, A.: From concept drift to model degradation: an overview on performance-aware drift detectors. Knowl.-Based Syst. **245**, 108632 (2022). https://doi.org/10.1016/j.knosys.2022.108632
21. Belkin, M., Hsu, D., Ma, S., Mandal, S.: Reconciling modern machine-learning practice and the classical bias-variance trade-off. Proc. Natl. Acad. Sci. **116**(32), 15849–15854 (2019). https://doi.org/10.1073/pnas.1903070116
22. Roelofs, R., Shankar, V., Recht, B., Fridovich-Keil, S., Hardt, M., Miller, J., Schmidt, L.: A meta-analysis of overfitting in machine learning. Neural Information Processing Systems, (2019). PDF URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/ee39e503b6bedf0c98c388b7e8589aca-Paper.pdf
23. Demšar, J., Zupan, B.: Hands-on training about overfitting. PLoS Comput. Biol. **17**(3), e1008671 (2021). https://doi.org/10.1371/journal.pcbi.1008671
24. Ying, X.: An overview of overfitting and its solutions. In Journal of Physics: Conference Series **1168**, 022022 (2019). https://doi.org/10.1088/1742-6596/1168/2/022022
25. Chicco, D.: Ten quick tips for machine learning in computational biology. BioData Mining **10**(1), 35 (2017). https://doi.org/10.1186/s13040-017-0155-3
26. Lee, B.D., Gitter, A., Greene, C.S., Raschka, S., Maguire, F., Titus, A.J., Kessler, M.D., Lee, A.J., Chevrette, M.G., Stewart, P.A., Britto-Borges, T., Cofer, E.M., Yu, K.-H., Carmona, J.J., Fertig, E.J., Kalinin, A.A., Signal, B., Lengerich, B.J., Triche, T.J., Jr., Boca, S.M.: Ten quick tips for deep learning in biology. PLoS Comput. Biol. **18**(3), e1009803 (2022). https://doi.org/10.1371/journal.pcbi.1009803
27. Chicco, D., Shiradkar, R.: Ten quick tips for computational analysis of medical images. PLoS Comput. Biol. **19**(1), e1010778 (2023). https://doi.org/10.1371/journal.pcbi.1010778
28. Bernett, J., Blumenthal, D.B., Grimm, D.G., Haselbeck, F., Joeres, R., Kalinina, O.V., List, M.: Guiding questions to avoid data leakage in biological machine learning applications. Nat. Methods **21**(8), 1444–1453 (2024). https://doi.org/10.1038/s41592-024-02285-2
29. Chicco, D., Jurman, G.: The ABC recommendations for validation of supervised machine learning results in biomedical sciences.

Frontiers in Big Data **5**, 979465 (2022). https://doi.org/10.3389/fdata.2022.979465
30. Wainberg, M., Alipanahi, B., Frey, B.J.: Are random forests truly the best classifiers? J. Mach. Learn. Res. **17**(110), 1–5 (2016). https://doi.org/10.5555/3045390.3045500
31. Fernández-Delgado, M., Cernadas, E., Barro, S., Amorim, D.: Do we need hundreds of classifiers to solve real world classification problems? J. Mach. Learn. Res. **15**(1), 3133–3181 (2014). https://doi.org/10.5555/2627435.2697065
32. Bushuiev, A., Bushuiev, R., Sedlar, J., Pluskal, T., Damborsky, J., Mazurenko, S., Sivic, J.: Revealing data leakage in protein interaction benchmarks. arXiv preprint arXiv:2404.10457, (2024). https://doi.org/10.48550/arXiv.2404.10457
33. Whalen, S., Truty, R.M., Pollard, K.S.: Enhancer-promoter interactions are encoded by complex genomic signatures on looping chromatin. Nat. Genet. **48**(5), 488–496 (2016). https://doi.org/10.1038/ng.3539
34. Cao, F., Fullwood, M.J.: Inflated performance measures in enhancer-promoter interaction-prediction methods. Nat. Genet. **51**(8), 1196–1198 (2019). https://doi.org/10.1038/s41588-019-0456-y
35. Whalen, S., Pollard, K.S.: Reply to 'inflated performance measures in enhancer-promoter interaction-prediction methods'. Nat. Genet. **51**(8), 1198–1200 (2019). https://doi.org/10.1038/s41588-019-0462-0
36. Xi, W., Beer, M.A.: Local epigenomic state cannot discriminate interacting and non-interacting enhancer-promoter pairs with high accuracy. PLoS Comput. Biol. **14**(12), e1006625 (2018). https://doi.org/10.1371/journal.pcbi.1006625
37. Schreiber, J., Singh, R., Bilmes, J., Noble, W.S.: A pitfall for machine learning methods aiming to predict across cell types. Genome Biol. **21**(1), 282 (2020). https://doi.org/10.1186/s13059-020-02216-z
38. Ilyas, I. F., Chu, X.: Data cleaning. Morgan & Claypool, San Rafael, California, USA, (2019). https://doi.org/10.2200/S00899ED1V01Y201909DTM057
39. García, S., Luengo, J., Herrera, F.: Data Preprocessing in Data Mining, Springer (2015). https://doi.org/10.1007/978-3-319-10247-4
40. Kazi, J.U.: Biomedical data preprocessing. In Python Essentials for Biomedical Data Analysis: An Introductory Textbook (2025). https://doi.org/10.1007/978-3-031-90072-4_2
41. Ampavathi, A., Saradhi, T.V.: Research challenges and future directions towards medical data processing. Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization **10**(6), 633–652 (2022). https://doi.org/10.1080/21681163.2022.2059411
42. Barrabés, M., Perera, M., Moriano, V.N., Giró-I-Nieto, X., Montserrat, D.M., Ioannidis, A.G.: Advances in biomedical missing data imputation: a survey. IEEE Access **13**, 16918–16932 (2024). https://doi.org/10.1109/ACCESS.2024.3480150
43. Rangayyan, D. M., Krishnan, S.: Biomedical signal analysis. John Wiley & Sons, Hoboken, New Jersey, USA, (2024). https://doi.org/10.1002/9781119893042
44. AlHinai, N.: Introduction to biomedical signal processing and artificial intelligence. In Biomedical Signal Processing and Artificial Intelligence in Healthcare (2020). https://doi.org/10.1016/B978-0-12-818946-7.00001-9
45. Eling, N., Morgan, M.D., Marioni, J.C.: Challenges in measuring and understanding biological noise. Nat. Rev. Genet. **20**(9), 536–548 (2019). https://doi.org/10.1038/s41576-019-0130-6
46. Scarciglia, A., Catrambone, V., Bonanno, C., Valenza, G.: Physiological noise: definition, estimation, and characterization in complex biomedical signals. IEEE Trans. Biomed. Eng. **71**(1), 45–55 (2023). https://doi.org/10.1109/TBME.2023.3285710
47. Sprang, M., Andrade-Navarro, M.A., Fontaine, J.F.: Batch effect detection and correction in RNA-seq data using machine-learning-based automated assessment of quality. BMC Bioinformatics **23**, 279 (2022). https://doi.org/10.1186/s12859-022-04825-9
48. Sun, H., Cui, Y., Wang, H., Liu, H., Wang, T.: Comparison of methods for the detection of outliers and associated biomarkers in mislabeled omics data. BMC Bioinformatics **21**(1), 357 (2020). https://doi.org/10.1186/s12859-020-03683-0
49. Liew, A.W.C., Law, N.F., Yan, H.: Missing value imputation for gene expression data: computational techniques to recover missing data from available information. Brief. Bioinform. **12**(5), 498–513 (2011). https://doi.org/10.1093/bib/bbq080
50. Tavazzi, E., Daberdaku, D., Vasta, R., Calvo, A., Chiò, A., Di Camillo, B.: Exploiting mutual information for the imputation of static and dynamic mixed-type clinical data with an adaptive *k*-nearest neighbours approach. BMC Med. Inform. Decis. Mak. **20**, 174 (2020). https://doi.org/10.1186/s12911-020-01196-1
51. Oneto, L., Chicco, D.: Eight quick tips for biologically and medically informed machine learning. PLoS Comput. Biol. **21**(1), e1012711 (2025). https://doi.org/10.1371/journal.pcbi.1012711
52. Leiser, F., Rank, S., Schmidt-Kraepelin, M., Thiebes, S., Sunyaev, A.: Medical informed machine learning: A scoping review and future research directions. Artif. Intell. Med. **145**, 102676 (2023). https://doi.org/10.1016/j.artmed.2023.102676
53. Shi, X., Prins, C., Van Pottelbergh, G., Mamouris, P., Vaes, B., De Moor, B.: An automated data cleaning method for electronic health records by incorporating clinical knowledge. BMC Med. Inform. Decis. Mak. **21**(1), 267 (2021). https://doi.org/10.1186/s12911-021-01651-4
54. Moreau, D., Wiebels, K.: Nine quick tips for open meta-analyses. PLoS Comput. Biol. **20**(7), e1012252 (2024). https://doi.org/10.1371/journal.pcbi.1012252
55. Foster, E.D., Deardorff, A.: Open science framework (OSF). J. Med. Libr. Assoc. **105**(2), 203 (2017). https://doi.org/10.5195/jmla.2017.88
56. Wang, Z., Wang, P., Liu, K., Wang, P., Fu, Y., Lu, C.T., Aggarwal, C.C., Pei, J., Zhou, Y.: A comprehensive survey on data augmentation. IEEE Trans. Knowl. Data Eng. **38**(1), 47–66 (2025). https://doi.org/10.1109/TKDE.2025.3531317
57. Goceri, E.: Medical image data augmentation: techniques, comparisons and interpretations. Artif. Intell. Rev. **56**(11), 12561–12605 (2023). https://doi.org/10.1007/s10462-023-10494-2
58. Marouf, M., Machart, P., Bansal, V., Kilian, C., Magruder, D.S., Krebs, C.F., Bonn, S.: Realistic in silico generation and augmentation of single-cell rna-seq data using generative adversarial networks. Nat. Commun. **11**(1), 166 (2020). https://doi.org/10.1038/s41467-019-14018-9
59. Albehairi, S., Khan, S., Alotaibi, R., Alganmi, N., Patwary, M.: Leveraging ai for biomedical data augmentation: a comparative review of model characteristics, performance analysis, and future research directions. International Journal of Data Science and Analytics **21**(1), 33 (2026). https://doi.org/10.1007/s41060-025-00642-5
60. Van Breugel, B., Liu, T., Oglic, D., Van Der Schaar, M.: Synthetic data in biomedicine via generative artificial intelligence. Nature Reviews Bioengineering **2**(12), 991–1004 (2024). https://doi.org/10.1038/s44222-024-00202-4
61. Holmes, J.H., Beinlich, J., Boland, M.R., Bowles, K.H., Chen, Y., Cook, T.S., Demiris, G., Draugelis, M., Fluharty, L., Gabriel, P.E., Grundmeier, R., Hanson, C.W., Herman, D.S., Himes, B.E., Hubbard, R.A., Kahn, C.E., Kim, D., Koppel, R., Long, Q., Mirkovic, N., Morris, J.S., Mowery, D.L., Ritchie, M.D., Urbanowicz, R., Moore, J.H.: Why is the electronic health record so challenging for research and clinical care? Methods Inf. Med. **60**(01/02), 032–048 (2021). https://doi.org/10.1055/s-0041-1731784

62. Chicco, D., Oneto, L., Tavazzi, E.: Eleven quick tips for data cleaning and feature engineering. PLoS Comput. Biol. **18**(12), e1010718 (2022). https://doi.org/10.1371/journal.pcbi.1010718
63. Ali, M.D.S., Islam, M.D.K., Das, A.A., Duranta, D.U.S., Haque, M.S.T.F., Rahman, M.D.H.: A novel approach for best parameters selection and feature engineering to analyze and detect diabetes: machine learning insights. Biomed. Res. Int. **2023**(1), 8583210 (2023). https://doi.org/10.1155/2023/8583210
64. Zeng, Z., Ma, Y., Hu, L., Tan, B., Liu, P., Wang, Y., Xing, C., Xiong, Y., Du, H.: OmicVerse: a framework for bridging and deepening insights across bulk and single-cell sequencing. Nat. Commun. **15**(1), 5983 (2024). https://doi.org/10.1038/s41467-024-50336-7
65. Pearson, K.: On lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science **2**(11), 559–572 (1901). https://doi.org/10.1080/14786440109462720
66. Van der Maaten, L., Hinton, G.: Visualizing data using *t*-SNE. J. Mach. Learn. Res. **9**(11), 2579–2605 (2008). https://doi.org/10.5555/2981780.2981833
67. Linderman, G.C., Rachh, M., Hoskins, J.G., Steinerberger, S., Kluger, Y.: Fast interpolation-based t-SNE for improved visualization of single-cell RNA-seq data. Nat. Methods **16**(3), 243–245 (2019). https://doi.org/10.1038/s41592-018-0308-4
68. Kobak, D., Berens, P.: The art of using *t*-SNE for single-cell transcriptomics. Nat. Commun. **10**(1), 5416 (2019). https://doi.org/10.1038/s41467-019-13056-x
69. McInnes, L., Healy, J., Melville, J.: UMAP: uniform manifold approximation and projection for dimension reduction. (2018). https://doi.org/10.48550/arXiv.1802.03426, arXiv preprint arXiv:1802.03426
70. Amid, E., Warmuth, M. K.: TriMap: large-scale dimensionality reduction using triplets. (2019). https://doi.org/10.48550/arXiv.1910.00204, arXiv preprint arXiv:1910.00204
71. Wang, Y., Huang, H., Rudin, C., Shaposhnik, Y.: Understanding how dimension reduction tools work: an empirical approach to deciphering *t*-SNE, UMAP, TriMAP, and PaCMAP for data visualization. J. Mach. Learn. Res. **22**(201), 1–73 (2021). https://doi.org/10.5555/3540261.3540468
72. Jacomy, M., Venturini, T., Heymann, S., Bastian, M.: ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software. PLoS ONE **9**(6), e98679 (2014). https://doi.org/10.1371/journal.pone.0098679
73. Moon, K.R., van Dijk, D., Wang, Z., Gigante, S., Burkhardt, D.B., Chen, W.S., Yim, K., van den Elzen, A., Hirn, M.J., Coifman, R.R., Ivanova, N.B., Wolf, G., Krishnaswamy, S.: Visualizing structure and transitions in high-dimensional biological data. Nat. Biotechnol. **37**(12), 1482–1492 (2019). https://doi.org/10.1038/s41587-019-0336-3
74. Jing, R., Xue, L., Li, M., Yu, L., Luo, J.: layerUMAP: a tool for visualizing and understanding deep learning models in biological sequence classification using UMAP. iScience (2022). https://doi.org/10.1016/j.isci.2022.105530
75. Kobak, D., Linderman, G.C.: Initialization is critical for preserving global data structure in both t-SNE and UMAP. Nat. Biotechnol. **39**(2), 156–157 (2021). https://doi.org/10.1038/s41587-020-00809-z
76. Jeon, H., Cho, A., Jang, J., Lee, S., Hyun, J., Ko, H.K., Jo, J., Seo, J.: Zadu: a Python library for evaluating the reliability of dimensionality reduction embeddings. In IEEE Visualization and Visual Analytics (2023). https://doi.org/10.1109/VIS54172.2023.00040
77. Chicco, D., Melzi, S., Gasparini, F., Jurman, G.: The advantages of our proposed Saturn coefficient over continuity and trustworthiness for UMAP dimensionality reduction evaluation. PeerJ Computer Science **12**, e3424 (2026). https://doi.org/10.7717/peerj-cs.3424
78. Hardt, M.: Climbing a shaky ladder: better adaptive risk estimation. (2017). https://doi.org/10.48550/arXiv.1706.02733, arXiv preprint arXiv:1706.02733
79. Shalev-Shwartz, S., Ben-David, S.: Understanding Machine Learning: from Theory to Algorithms, Cambridge University Press, Cambridge, England, United Kingdom (2014). https://doi.org/10.1017/CBO9781107298019
80. C. M. Bishop and H. Bishop. Deep Learning: Foundations and Concepts. Springer Nature, Cham, Switzerland, 2023. https://doi.org/10.1007/978-3-031-45468-4
81. Zhong, Y., Chalise, P., He, J.: Nested cross-validation with ensemble feature selection and classification model for high-dimensional biological data. Communications in Statistics-Simulation and Computation **52**(1), 110–125 (2023). https://doi.org/10.1080/03610918.2021.1960382
82. Oneto, L.: Model Selection and Error Estimation in a Nutshell, Springer (2020). https://doi.org/10.1007/978-3-030-41611-6
83. Singh, K., Woodward, M.A.: The rigorous work of evaluating consistency and accuracy in electronic health record data. JAMA Ophthalmology **139**(8), 894–895 (2021). https://doi.org/10.1001/jamaophthalmol.2021.1779
84. Penev, Y.P., Buchanan, T.R., Ruppert, M.M., Liu, M., Shekouhi, R., Guan, Z., Balch, J., Ozrazgat-Baslanti, T., Shickel, B., Loftus, T.J., Bihorac, A.: Electronic health record data quality and performance assessments: scoping review. JMIR Med. Inform. **12**(1), e58130 (2024). https://doi.org/10.2196/58130
85. Samadi, M.E., Mirzaieazar, H., Mitsos, A., Schuppert, A.: Noisecut: a Python package for noise-tolerant classification of binary data using prior knowledge integration and max-cut solutions. BMC Bioinform. (2024). https://doi.org/10.1186/s12859-024-05769-8
86. Oneto, L., Chicco, D.: Nine quick tips for trustworthy machine learning in the biomedical sciences. PLoS Comput. Biol. **21**(10), e1013624 (2025). https://doi.org/10.1371/journal.pcbi.1013624
87. Probst, P., Boulesteix, A.-L., Bischl, B.: Tunability: importance of hyperparameters of machine learning algorithms. J. Mach. Learn. Res. **20**(53), 1–32 (2019). https://doi.org/10.5555/3454287.3454336
88. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001). https://doi.org/10.1023/A:1010933404324
89. Al-Janabi, M., Andras, P.: A systematic analysis of random forest based social media spam classification. In International Conference on Network and System Security (2017). https://doi.org/10.1007/978-3-319-64701-2_27
90. Bilolikar, D.K., More, A., Gong, A., Janssen, J.: How to outperform default random forest regression: choosing hyperparameters for applications in large-sample hydrology. Hydrol. Res. **57**(1), 61–77 (2026). https://doi.org/10.2166/nh.2025.046
91. Braga-Neto, U.M., Dougherty, E.R.: Is cross-validation valid for small-sample microarray classification? Bioinformatics **20**(3), 374–380 (2004). https://doi.org/10.1093/bioinformatics/btg419
92. Kleiner, A., Talwalkar, A., Sarkar, P., Jordan, M.I.: A scalable bootstrap for massive data. J. R. Stat. Soc. Ser. B Stat Methodol. **76**(4), 795–816 (2014). https://doi.org/10.1111/rssb.12050
93. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. J. Mach. Learn. Res. **13**(1), 281–305 (2012). https://doi.org/10.5555/2188385.2188395
94. Wu, J., Chen, X.Y., Zhang, H., Xiong, L.D., Lei, H., Deng, S.H.: Hyperparameter optimization for machine learning models based on bayesian optimization. Journal of Electronic Science and Technology **17**(1), 26–40 (2019). https://doi.org/10.11989/JEST.1674-862X.80904120
95. Young, S.R., Rose, D.C., Karnowski, T.P., Lim, S.H., Patton, R.M.: Optimizing deep learning hyper-parameters through an evolutionary algorithm. In Machine Learning in High-

performance Computing Environments (2015). https://doi.org/10.1145/2834892.2834896
96. Maclaurin, D., Duvenaud, D., Adams, R.: Gradient-based hyperparameter optimization through reversible learning. In International Conference on Machine Learning, (2015). https://doi.org/10.48550/arXiv.1502.03492
97. Jomaa, H. S., Grabocka, J., Schmidt-Thieme, L.: Hyp-RL: hyperparameter optimization by reinforcement learning. (2019). https://doi.org/10.48550/arXiv.1906.11527, arXiv preprint arXiv:1906.11527,
98. Ren, P., Xiao, Y., Chang, X., Huang, P.Y., Li, Z., Chen, X., Wang, X.: A comprehensive survey of neural architecture search: challenges and solutions. ACM Comput. Surv. **54**(4), 1–34 (2021). https://doi.org/10.1145/3447582
99. Dwork, C., Feldman, V., Hardt, M., Pitassi, T., Reingold, O., Roth, A.: The reusable holdout: preserving validity in adaptive data analysis. Science **349**(6248), 636–638 (2015). https://doi.org/10.1126/science.aaa9375
100. Chicco, D., Jurman, G.: The Matthews correlation coefficient (MCC) should replace the ROC AUC as the standard metric for assessing binary classification. BioData Mining **16**(1), 4 (2023). https://doi.org/10.1186/s13040-023-00322-4
101. Chicco, D., Jurman, G.: An invitation to greater use of Matthews correlation coefficient in robotics and artificial intelligence. Frontiers in Robotics and AI **9**, 876814 (2022). https://doi.org/10.3389/frobt.2022.876814
102. Chicco, D., Warrens, M.J., Jurman, G.: The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. PeerJ Computer Science **7**, e623 (2021). https://doi.org/10.7717/peerj-cs.623
103. Giudici, P.: Safe machine learning. Statistics **58**(3), 473–477 (2024). https://doi.org/10.1080/02331888.2024.2361481
104. Giudici, P., Centurelli, M., Turchetta, S.: Artificial intelligence risk measurement. Expert Syst. Appl. **235**, 121220 (2024). https://doi.org/10.1016/j.eswa.2023.121220
105. Babaei, G., Giudici, P., Raffinetti, E.: A rank graduation box for SAFE AI. Expert Syst. Appl. **259**, 125239 (2025). https://doi.org/10.1016/j.eswa.2024.125239
106. Giudici, P., Kolesnikov, V.: SAFE AI metrics: An integrated approach. Machine Learning with Applications **23**, 100821 (2026). https://doi.org/10.1016/j.mlwa.2025.100821
107. Spelta, A., Raffinetti, E.: Evaluating SAFE AI principles using wasserstein distance: a comparative study of machine learning models. Statistics **58**(6), 1283–1303 (2024). https://doi.org/10.1080/02331888.2024.2419420
108. Giudici, P., Raffinetti, E.: RGA: a unified measure of predictive accuracy. Adv. Data Anal. Classif. **19**(1), 67–93 (2025). https://doi.org/10.1007/s11634-023-00574-2
109. Chicco, D., Jurman, G.: The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics **21**(1), 6 (2020). https://doi.org/10.1186/s12864-019-6413-7
110. Yao, Y., Rosasco, L., Caponnetto, A.: On early stopping in gradient descent learning. Constr. Approx. **26**(2), 289–315 (2007). https://doi.org/10.1007/s00365-006-0663-2
111. Aburass, S., Rumman, M.A.: Quantifying overfitting: introducing the overfitting index. In: International Conference on Electrical, Computer and Energy Technologies (2024). https://doi.org/10.1109/ICECET61426.2024.10698796
112. Hawkins, D.M.: The problem of overfitting. J. Chem. Inf. Comput. Sci. **44**(1), 1–12 (2004). https://doi.org/10.1021/ci0342472
113. Mersha, M., Lam, K., Wood, J., Alshami, A.K., Kalita, J.: Explainable artificial intelligence: a survey of needs, techniques, applications, and future direction. Neurocomputing **599**, 128111 (2024). https://doi.org/10.1016/j.neucom.2024.128111
114. Budhkar, A., Song, Q., Su, J., Zhang, X.: Demystifying the black box: A survey on explainable artificial intelligence (XAI) in bioinformatics. Comput. Struct. Biotechnol. J. **27**, 346–359 (2025). https://doi.org/10.1016/j.csbj.2024.12.027
115. Lapuschkin, S., Waldchen, S., Binder, A., Montavon, G., Samek, W., Muller, K.-R.: Unmasking clever hans predictors and assessing what machines really learn. Nat. Commun. **10**(1), 1096 (2019). https://doi.org/10.1038/s41467-019-08987-4
116. DeGrave, A.J., Janizek, J.D., Lee, S.-I.: AI for radiographic COVID-19 detection selects shortcuts over signal. Nature Machine Intelligence **3**(7), 610–619 (2021). https://doi.org/10.1038/s42256-021-00338-7
117. Giudici, P., Raffinetti, E.: Shapley-Lorenz explainable artificial intelligence. Expert Syst. Appl. **167**, 114104 (2021). https://doi.org/10.1016/j.eswa.2020.114104
118. Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., Kim, B.: Sanity checks for saliency maps. In Advances in Neural Information Processing systems, (2018). https://doi.org/10.48550/arXiv.1810.03292
119. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature machine intelligence **1**(5), 206–215 (2019). https://doi.org/10.1038/s42256-019-0048-x
120. Sheridan, R.P.: Time-split cross-validation as a method for estimating the goodness of prospective prediction. J. Chem. Inf. Model. **53**(4), 783–790 (2013). https://doi.org/10.1021/ci400084k
121. Wu, Y., Wershof, E., Schmon, S. M., Nassar, M., Osiński, B., Eksi, R., Yan, Z., Stark, R., Zhang, K., Graepel, T.: PerturBench: benchmarking machine learning models for cellular perturbation analysis. (2024). https://doi.org/10.48550/arXiv.2408.10609, arXiv preprint arXiv:2408.10609
122. Wong, E., Kolter, J. Z.: Learning perturbation sets for robust machine learning. (2020). https://doi.org/10.48550/arXiv.2007.08450, arXiv preprint arXiv:2007.08450
123. Yadav, P., Steinbach, M., Kumar, V., Simon, G.: Mining electronic health records (EHRs): a survey. ACM Comput. Surv. **50**(6), 1–40 (2018). https://doi.org/10.1145/3127881
124. D'Amour, A., Heller, K., Moldovan, D., Adlam, B., Alipanahi, B., Beutel, A., Chen, C., Deaton, J., Eisenstein, J., Hoffman, M.D., Hormozdiari, F., Houlsby, N., Hou, S., Jerfel, G., Karthikesalingam, A., Lucic, M., Ma, Y., McLean, C., Mincu, D., Mitani, A., Montanari, A., Nado, Z., Natarajan, V., Nielson, C., Osborne, T.F., Raman, R., Ramasamy, K., Sayres, R., Schrouff, J., Seneviratne, M., Sequeira, S., Suresh, H., Veitch, V., Vladymyrov, M., Wang, X., Webster, K., Yadlowsky, S., Yun, T., Zhai, X., Sculley, D.: Underspecification presents challenges for credibility in modern machine learning. J. Mach. Learn. Res. **23**(226), 1–61 (2022). https://doi.org/10.48550/arXiv.2011.03395
125. Chicco, D., Cerono, G., Cangelosi, D.: A survey on publicly available open datasets derived from electronic health records (EHRs) of patients with neuroblastoma. Data Science Journal **21**(1), 17–17 (2022). https://doi.org/10.5334/dsj-2022-017
126. Kustra, R., Zagdanski, A.: Data-fusion in clustering microarray data: balancing discovery and interpretability. IEEE/ACM Trans. Comput. Biol. Bioinf. **7**(1), 50–63 (2008). https://doi.org/10.1109/TCBB.2008.94
127. The Gene Ontology Consortium: The Gene Ontology knowledgebase in 2026. Nucleic Acids Res. **54**(D1), D1779–D1792 (2026). https://doi.org/10.1093/nar/gkaf1292
128. Pinoli, P., Chicco, D., Masseroli, M.: Computational algorithms to predict Gene Ontology annotations. BMC Bioinformatics **16**(Suppl 6), S4 (2015). https://doi.org/10.1186/1471-2105-16-S6-S4

129. Breitkreutz, B.-J., Stark, C., Tyers, M.: The GRID: the general repository for interaction datasets. Genome Biol. **4**(3), R23 (2003). https://doi.org/10.1186/gb-2003-4-3-r23
130. scikit-learn developers. Underfitting vs. overfitting. https://scikit-learn.org/stable/auto_examples/model_selection/plot_underfitting_overfitting.html, (2024). Accessed: 2026-04-10
131. Barry, M.: Probability of backtest overfitting. https://cran.r-project.org/web/packages/pbo/vignettes/pbo.html, (2022). R package vignette for the PBO package (version 1.3.5)
132. Akalin, A.: Model tuning and avoiding overfitting. https://compgenomr.github.io/book/model-tuning-and-avoiding-overfitting.html, (2020). Chapter 5.8 of "Computational genomics with R"; accessed 2026-04-10
133. Kurz, A. J.: Overfitting, regularization, and information criteria. https://bookdown.org/ajkurz/Statistical_Rethinking_recoded/overfitting-regularization-and-information-criteria.html, (2020). Chapter 6 of "Statistical rethinking with brms, ggplot2, and the tidyverse"; accessed 2026-04-10
134. Chollet, F., Kalinowski, T., Allaire, J. J.: Overfit and underfit. https://tensorflow.rstudio.com/tutorials/keras/overfit_and_underfit, (2023). TensorFlow for R (Keras) tutorial; accessed 2026-04-10
135. : Chiu, W.: Measuring overfitting. http://rpubs.com/crossxwill/measuring_overfitting, (2015). Accessed: 2026-04-10
136. Domingos, P.: A few useful things to know about machine learning. Commun. ACM **55**(10), 78–87 (2012). https://doi.org/10.1145/2347736.2347755