International Journal of Innovative Scientific & Engineering Technologies Research 14(3):224-240, July-Sept, 2026

© SEAHI PUBLICATIONS, 2026 www.seahipublications.org ISSN: 2360-896X

DOI:10.5281/zenodo.22934184

Concepts And Classification Of Machine Learning Algorithms: A Critical Review And Two-Axis Taxonomy

Timinipre Amabie*; Laeticia Nneka Onyejegbu; Barilee Baridam

Department of Computer Science,
Faculty of Computing,
University of Port Harcourt, Rivers State, Nigeria
***Corresponding Author: amabietimi@gmail.com**

ABSTRACT

The proliferation of machine-learning terminology has encouraged taxonomies that mix learning paradigms, prediction tasks, algorithm families, and model-control techniques. This critical narrative review clarifies these levels and proposes a two-axis taxonomy of machine-learning systems. Methods: Seminal and primary peer-reviewed literature was prioritized for definitions, algorithm origins, validation principles, and empirical examples; no new empirical dataset was generated. Learning paradigms are most clearly separated into supervised, unsupervised, semi-supervised, self-supervised, and reinforcement learning, while tasks include classification, regression, clustering, anomaly detection, association/frequent-pattern mining, dimensionality reduction, and generative modelling. Algorithms map many-to-many across these axes. Regularization, cross-validation, calibration, leakage control, interpretability, privacy, fairness, and deployment monitoring are cross-cutting procedures rather than task classes. Published primary studies illustrate both the capabilities of ML and the dependence of performance on datasets, reference standards, thresholds, and evaluation design. A multi-dimensional taxonomy improves technical accuracy, method selection, teaching, and reproducibility by distinguishing how a system learns from what it is designed to do and how it is evaluated.

Keywords: artificial intelligence; machine learning; supervised learning; unsupervised learning; classification; regression; model validation; data leakage

1. INTRODUCTION

Machine learning (ML) is best understood as a family of computational and statistical approaches in which performance on a task is improved through experience represented by data, rather than through a complete set of hand-coded rules. The intellectual lineage of the field predates the present use of the term: Turing (1950) framed machine intelligence as an operational question about observable behaviour, while Samuel (1959) described a checkers program that improved through experience. Later learning theory formalized the problem of learning functions that generalize beyond observed examples (Valiant, 1984). These foundations make one point especially important for contemporary definitions: machine learning does not require a sentient mind, intention, or human-like understanding; it requires a specified learning problem, data or interaction, an objective, and a procedure for updating a model.

Modern ML sits at the intersection of computer science, statistics, optimization, and domain science. Statistical learning emphasizes estimation, uncertainty, and generalization, whereas much of contemporary ML emphasizes scalable prediction, representation learning, and algorithmic performance.

Breiman (2001b) famously contrasted data models and algorithmic models, while Shmueli (2010) distinguished explanatory from predictive goals. Neither distinction implies that one tradition supersedes the other. Rather, model choice should follow the scientific or operational objective, the data-generating context, and the consequences of error (Jordan & Mitchell, 2015).

The rapid expansion of ML has produced a vocabulary problem. Terms such as supervised learning, classification, random forest, deep learning, regularization, anomaly detection, and reinforcement learning are often placed in a single list as if they describe the same type of thing. They do not. Supervised learning describes how a learning signal is supplied; classification describes a type of prediction task; random forest names an algorithmic family; and regularization describes a strategy for controlling model complexity. Conflating these levels makes algorithm selection harder and can lead to technically incorrect statements about what a method can do.

This review therefore has four objectives. First, it clarifies the core concepts required to describe an ML system precisely. Second, it proposes a two-axis taxonomy that separates learning paradigms from tasks or functions. Third, it maps representative algorithm families to those axes and explains why the mapping is many-to-many rather than one-to-one. Fourth, it connects taxonomy to model development, evaluation, applications, and limitations, with particular attention to data leakage, calibration, fairness, interpretability, privacy, robustness, and deployment. The emphasis is on original and seminal literature where possible, supplemented by peer-reviewed syntheses when they provide broader context.

2. Review Scope and Source Strategy

This article is a critical narrative review and conceptual synthesis, not a systematic review or meta-analysis. Literature was identified through targeted scholarly searches and publisher, journal, and conference-proceedings records using combinations of terms relating to machine learning definitions, learning paradigms, classification, regression, clustering, anomaly detection, association-rule mining, dimensionality reduction, regularization, validation, leakage, fairness, interpretability, privacy, security, and landmark applications. Priority was given to original papers that introduced or established methods (for example, support-vector machines, random forests, Apriori, DBSCAN, local outlier factor, isolation forest, LASSO, and Q-learning) and to primary empirical studies when numerical performance was discussed. Bibliographic details were checked against DOI, publisher, journal, or conference-proceedings records, and numerical statements were checked against the cited primary publications.

The review did not generate a new empirical dataset. Quantitative values in Table 4 are concise reports of results from the cited primary studies. Because those studies use different datasets, reference standards, task definitions, and evaluation protocols, their results are presented as study-specific illustrations and are neither pooled nor ranked across studies.

Because the purpose is conceptual clarification rather than exhaustive evidence synthesis, the search was targeted rather than protocol-driven, and no claim is made that it retrieved every relevant paper. Sources were included when they directly supported a definitional, taxonomic, methodological, evaluative, or application claim. Original method papers and peer-reviewed primary studies were preferred; peer-reviewed syntheses were used when broader context was required.

3. Conceptual Foundations of Machine Learning Systems

Artificial intelligence (AI) is the broader field concerned with constructing systems that perform tasks associated with intelligent behaviour; machine learning is a major approach within AI in which system behaviour is learned from data or interaction. Deep learning is a subset of ML based on multi-layer representation learning and neural networks (LeCun et al., 2015). The resurgence of deep learning was enabled by advances in optimization, larger datasets, and computing, building on earlier work on back-propagation (Rumelhart et al., 1986) and deep representation learning (Hinton & Salakhutdinov, 2006). Thus, AI, ML, and deep learning are nested but not interchangeable terms.

A dataset is a collection of observations described by variables or features. In supervised prediction, each training observation usually contains inputs X and a target y. The target may be a categorical label, as in

classification, or a numeric quantity, as in regression. Unlabelled data contain the inputs without externally supplied target values. Mixed datasets are central to semi-supervised learning, where a relatively small labelled subset is combined with a larger unlabelled set (Blum & Mitchell, 1998; Sohn et al., 2020). Self-supervised methods differ because supervision signals are constructed from the data themselves, such as predicting masked tokens or learning invariance between transformed views (Chen et al., 2020; Devlin et al., 2019).

The term ground truth should be used cautiously. A label may serve as the operational reference standard for training, but it can still contain measurement error, disagreement, historical bias, or proxy variables. This matters because algorithms optimize patterns in the provided data, not an abstract notion of truth. The health-care bias documented by Obermeyer et al. (2019), for example, arose because a widely used algorithm employed health-care cost as a proxy for health need; unequal access and expenditure patterns caused the proxy to encode racial inequity.

Generalization is the ability of a fitted model to perform adequately on relevant observations not used to fit or tune it. Generalization is distinct from memorization. A model can have very low training error and still fail on new data when it overfits noise, idiosyncratic correlations, or artifacts. Conversely, underfitting occurs when the model or representation is too restricted to capture important signal. Rigorous evaluation therefore requires separation between data used for model development and data used for final assessment. Cross-validation is useful for estimating model performance and tuning, but if model selection and performance estimation reuse the same information naively, the reported error can be optimistically biased (Cawley & Talbot, 2010; Kohavi, 1995; Varma & Simon, 2006).

An ML algorithm is a procedure for fitting or updating a model; a model is the fitted mathematical or computational representation produced by that procedure; and an ML system includes the model plus the data pipeline, preprocessing, software, decision thresholds, interfaces, monitoring, and human or organizational processes surrounding deployment. This distinction is important because many failures attributed to an 'algorithm' actually arise from the system around it, including data collection, feature leakage, feedback loops, calibration, or changing deployment conditions (Sculley et al., 2015).

4. A Two-Axis Classification of Machine Learning Systems

A useful classification must answer two different questions: (1) how does the system obtain a learning signal, and (2) what computational or predictive task is being performed? Figure 1 visualizes this separation and shows that regularization, evaluation, and deployment controls cut across both axes.

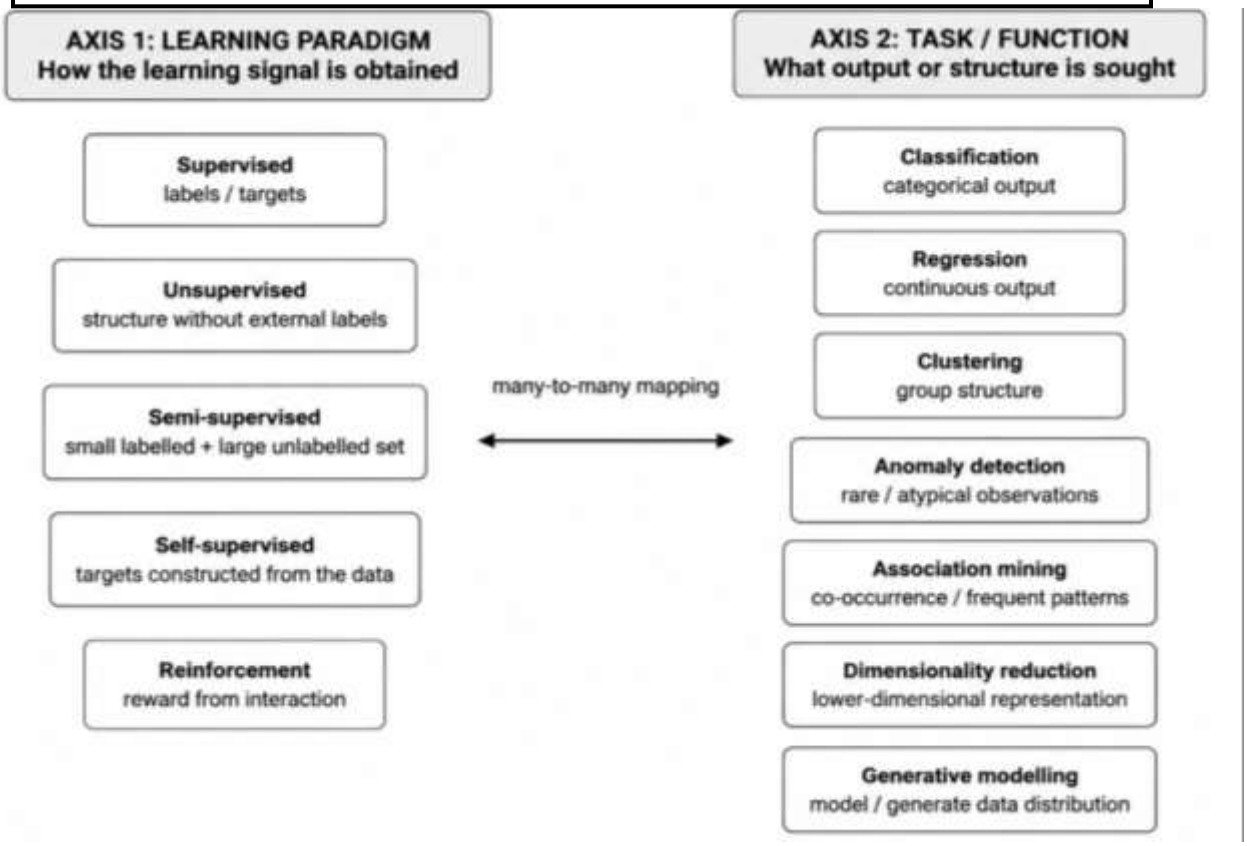

Figure 1. Two-axis taxonomy of machine learning systems. ***Note.*** **The figure separates learning paradigm (source of the learning signal) from task/function (desired output). Algorithms and architectures map many-to-many across the two axes. Developed from the synthesis in this review, drawing on foundational work cited in Sections 4.1–4.2.**

4.1 Axis 1: Classification by Learning Paradigm

The first axis of the proposed taxonomy is the learning paradigm: the source and form of the learning signal. Figure 1 separates this axis from the task or function the model performs. The separation resolves a recurrent ambiguity in introductory classifications because the same learning paradigm can support several tasks, and the same task can be approached under different paradigms.

Supervised learning uses paired inputs and target values to estimate a mapping that can be applied to new observations whose targets are unknown at prediction time. Labels are required for supervised training and evaluation, whereas the fitted model accepts previously unseen inputs for which targets are not yet known. Representative supervised algorithms include logistic regression, decision trees, random forests, support-vector machines, k-nearest neighbours, gradient-boosted models, and neural networks (Cortes & Vapnik, 1995; Cover & Hart, 1967; Friedman, 2001; Quinlan, 1986; Breiman, 2001a).

Unsupervised learning seeks useful structure in data without externally supplied target labels. Typical objectives include clustering, density or manifold estimation, latent representation learning, and some forms of anomaly detection. K-means partitions observations around centroids (MacQueen, 1967), DBSCAN identifies density-connected clusters and can label low-density points as noise (Ester et al., 1996), and principal-component or manifold methods find lower-dimensional representations (Pearson, 1901; Tenenbaum et al., 2000). Unsupervised learning is not inherently 'less accurate' than supervised learning; accuracy against class labels is often not even the relevant objective. Its evaluation must instead match the unsupervised task, such as cluster stability, reconstruction quality, density fit, or downstream usefulness.

Semi-supervised learning combines labelled and unlabelled observations. Co-training demonstrated how two sufficiently informative views of the data could allow classifiers to label additional examples iteratively (Blum & Mitchell, 1998). Modern methods such as FixMatch combine pseudo-labeling with consistency regularization and data augmentation; Sohn et al. (2020) reported strong CIFAR-10 performance with very small labelled subsets. The key concept is not that the labelled portion simply 'labels' all remaining observations with certainty, but that the learning objective exploits both labelled supervision and structural information in unlabelled data while managing uncertainty.

Self-supervised learning constructs a pretext or predictive objective directly from unlabelled data. Contrastive learning can train representations by bringing related transformed views closer while separating unrelated examples (Chen et al., 2020). In language, BERT was pre-trained with masked-token and related objectives before supervised fine-tuning for downstream tasks (Devlin et al., 2019). Self-supervision is therefore best treated as its own learning paradigm or as a major branch of representation learning rather than being collapsed into conventional unsupervised learning.

Reinforcement learning (RL) is based on sequential interaction. An agent observes a state, chooses an action, receives a reward, and updates a policy or value estimate to improve expected cumulative reward. Temporal-difference learning (Sutton, 1988), Q-learning (Watkins & Dayan, 1992), and REINFORCE-style policy gradients (Williams, 1992) are foundational approaches. Deep RL later combined neural representation learning with RL, exemplified by deep Q-networks for Atari games (Mnih et al., 2015) and the policy/value networks and tree search used in AlphaGo (Silver et al., 2016). RL is therefore not defined by a particular neural architecture but by the interaction-and-reward learning problem.

Table 1 condenses the distinctions. Its most important implication is that the presence or absence of labels characterizes training information, not whether the trained model can accept a new unlabelled input at inference time.

Table 1. Core distinctions among major learning paradigms

Learning paradigm	Training signal	Primary objective	Typical tasks	Representative methods	Key clarification
Supervised	Externally supplied labels/targets for training	Predict target for new inputs	Classification; regression; ranking	Logistic regression; decision trees; random forests; SVM; k-NN; boosting; neural networks	Unseen prediction inputs need not be labelled; the labels are needed to learn/evaluate the mapping.
Unsupervised	No external target labels	Discover structure or representations	Clustering; density estimation; dimensionality reduction; some anomaly detection	K-means; DBSCAN; PCA; NMF; Isomap; t-SNE	Evaluation depends on the structural objective; supervised accuracy is often inappropriate.
Semi-supervised	Small labelled set + larger unlabelled set	Improve predictive learning with limited labels	Classification; regression; representation learning	Co-training; pseudo-labeling; consistency methods such as FixMatch	Pseudo-labels are uncertain model outputs, not automatically equivalent to ground truth.
Self-supervised	Targets generated from the data or transformations	Learn transferable representations	Representation learning; pretraining for language, vision and multimodal tasks	Contrastive learning; masked prediction; transformer pretraining	Usually followed by fine-tuning, probing, or downstream evaluation.
Reinforcement learning	Rewards obtained through sequential interaction	Learn policy/value function to maximize cumulative reward	Control; planning; sequential decision making	Temporal-difference learning; Q-learning; policy gradients; deep RL	Requires careful treatment of exploration, delayed rewards, safety and environment assumptions.

4.2 Axis 2: Classification by Task or Function

The second axis describes the task or functional objective. Task categories concern the type of output or structure sought, not the source of the learning signal. Table 2 summarizes the major task families and

distinguishes categories that are often conflated when tasks, algorithms, and model-control techniques are treated as equivalent.

Classification assigns inputs to discrete categories. Binary classification has two classes; multiclass classification has more than two mutually exclusive classes; multilabel classification can assign several labels simultaneously. Logistic regression is a classification model because it models class probability for a categorical outcome, despite the word 'regression' in its name. Other classification methods include decision trees (Quinlan, 1986), random forests (Breiman, 2001a), support-vector machines (Cortes & Vapnik, 1995), k-nearest neighbours (Cover & Hart, 1967), boosted classifiers (Freund & Schapire, 1997), and neural networks.

Regression predicts a continuous or otherwise quantitative response. Ordinary least-squares linear regression is the classical baseline; ridge regression adds an L2 penalty to shrink coefficients (Hoerl & Kennard, 1970), LASSO uses an L1 penalty that can set coefficients exactly to zero (Tibshirani, 1996), and elastic net combines L1 and L2 penalties (Zou & Hastie, 2005). Nonlinear alternatives include regression trees and forests, support-vector regression, locally weighted regression (Cleveland, 1979), multivariate adaptive regression splines (Friedman, 1991), and gradient boosting (Friedman, 2001). Least-angle regression provides an efficient path-building approach closely related to shrinkage and variable selection (Efron et al., 2004). The defining property is the outcome and loss formulation, not whether the underlying algorithm is 'statistical' or 'machine learning'.

Clustering groups observations according to similarity, density, or another structural criterion without requiring predefined class labels. K-means is centroid-based and favors approximately compact clusters, while DBSCAN is density-based and can identify irregularly shaped clusters and noise points (Ester et al., 1996; MacQueen, 1967). Neural networks are not, by themselves, a clustering algorithm; they may instead learn representations that are subsequently clustered or be embedded within specialized clustering architectures.

Anomaly detection identifies observations, events, or patterns that are atypical relative to a reference distribution or neighbourhood. Local Outlier Factor compares local density with that of neighbours (Breunig et al., 2000); one-class support-vector methods estimate the support of a high-dimensional distribution (Schölkopf et al., 2001); and isolation forest identifies anomalies through unusually short isolation paths in randomized trees (Liu et al., 2008). Simple z-scores can be useful statistical screening rules, but they should not be presented as a general ML algorithm, and their normality assumptions may be inappropriate for many anomaly problems.

Association-rule and frequent-pattern mining identify frequent itemsets and rules that satisfy defined support, confidence, or other interestingness thresholds. The foundational work of Agrawal et al. (1993) and Agrawal and Srikant (1994) established efficient discovery of frequent itemsets and association rules; FP-growth later avoided candidate generation by using a compressed frequent-pattern tree (Han et al., 2000). Isolation Forest and Local Outlier Factor are anomaly-detection methods (Breunig et al., 2000; Liu et al., 2008), whereas z-scores are statistical screening rules; none is an association-rule algorithm.

Dimensionality reduction seeks a lower-dimensional representation while preserving information relevant to a defined criterion. Principal component analysis (PCA) finds orthogonal directions of maximal variance (Pearson, 1901); non-negative matrix factorization produces additive parts-based representations under non-negativity constraints (Lee & Seung, 1999); Isomap preserves approximate geodesic distances on a manifold (Tenenbaum et al., 2000); and t-SNE is primarily a visualization technique that emphasizes local neighbourhood structure in low dimensions (van der Maaten & Hinton, 2008). Deep autoencoder approaches can also learn nonlinear low-dimensional codes (Hinton & Salakhutdinov, 2006). Dimensionality reduction can improve visualization or computational efficiency, but it can also discard predictive signal if selected without regard to the downstream objective.

Generative modelling learns aspects of the data distribution so that new samples, latent representations, or conditional outputs can be generated. Generative adversarial networks (GANs) train a generator against a discriminator in a minimax game (Goodfellow et al., 2014). Transformer architectures, introduced for sequence transduction by Vaswani et al. (2017), later became a central foundation for large-scale self-supervised language modelling. Generative modelling is included here because a modern taxonomy that stops at classification, regression, clustering, and dimensionality reduction omits a major class of contemporary ML objectives.

Regularization should not be listed as a functional task. It is a cross-cutting model-control strategy used to reduce variance, constrain complexity, or encourage desirable parameter structure. Ridge, LASSO, and elastic net are penalized estimators (Hoerl & Kennard, 1970; Tibshirani, 1996; Zou & Hastie, 2005), while dropout regularizes neural networks through stochastic omission of units during training (Srivastava et al., 2014). Regularization can be used within classification, regression, representation learning, and other tasks; its role is therefore orthogonal to the task taxonomy.

As summarized in Table 2, task categories are defined by the output and objective. Evaluation must also be task-specific: classification metrics are not automatically valid for clustering, and a dimensionality-reduction method is not evaluated simply by the accuracy of a label it was never asked to predict.

Table 2. Machine-learning tasks, objectives, representative methods and evaluation measures

Task / function	Output	Typical objective / loss	Representative methods	Common evaluation
Classification	Categorical label or probability	Log loss; hinge loss; impurity measures	Logistic regression; trees/forests; SVM; k-NN; boosting; neural networks	Accuracy, sensitivity/recall, specificity, precision, F1, ROC-AUC, PR-AUC, calibration
Regression	Continuous numeric value	Squared, absolute, Huber, quantile or task-specific loss	Linear/ridge/LASSO/elastic net; trees/forests; SVR; MARS; boosting; neural networks	MAE, RMSE/MSE, R^2, calibration/coverage when probabilistic
Clustering	Cluster membership / structure	Within-cluster dispersion, density or graph/manifold objective	K-means; DBSCAN; hierarchical methods	Silhouette, stability, external indices when reference labels exist
Anomaly detection	Anomaly score / binary flag	Density, distance, isolation or reconstruction objective	LOF; one-class SVM; isolation forest; autoencoders	Precision/recall at relevant threshold; PR-AUC; false-alarm rate; time-to-detection
Association / frequent-pattern mining	Frequent itemsets and rules	Support, confidence, lift and related interestingness criteria	Apriori; FP-growth	Support, confidence, lift; domain utility; validation on held-out transactions
Dimensionality reduction	Low-dimensional coordinates / components	Variance, reconstruction, neighbourhood or manifold objective	PCA; NMF; Isomap; t-SNE; autoencoders	Explained variance, reconstruction error, neighbourhood preservation, downstream performance
Generative modelling	Generated samples / conditional distributions	Likelihood, adversarial, reconstruction or denoising objectives	GANs; autoregressive / transformer models; variational families	Task-specific fidelity, diversity, likelihood/perplexity, human/domain evaluation

Note. Metrics are illustrative and must be chosen according to class prevalence, decision costs, data structure, and the intended use of the model.

5. Mapping Algorithm Families Across Paradigms and Tasks

Table 3 makes the many-to-many relationship explicit. An algorithm family is not permanently attached to a single task. Decision trees and random forests can perform classification or regression; support-vector methods have classification, regression, and one-class variants; neural networks can be configured for classification, regression, representation learning, generation, anomaly detection, or reinforcement learning. Conversely, a single task such as classification can be solved by many algorithm families. Algorithm selection should therefore follow the data type, sample size, target, loss, interpretability requirements, computational constraints, and expected deployment conditions rather than a one-dimensional taxonomy.

This also explains why lists of 'classification algorithms' and 'regression algorithms' can overlap. The overlap is not an error when the algorithm family genuinely supports different objective functions. The error occurs when the task, paradigm, and technique are not distinguished. For example, random forest is an algorithmic family, supervised learning is a paradigm, classification is a task, and cross-validation is an evaluation procedure. Treating all four as peer categories obscures the design logic of an ML system.

There is also no general law that a more complex method is more accurate. Simple models can outperform complex models when the signal-to-noise ratio, sample size, or data structure favors them, and complex models can overfit if selection is not carefully nested within validation. Comparative evaluation should therefore include credible baselines and preserve a clean estimate of out-of-sample performance (Breiman, 2001b; Cawley & Talbot, 2010; Domingos, 2012; Shmueli, 2010).

Table 3. Many-to-many mapping between representative algorithm families, learning paradigms and tasks

Algorithm / model family	Common paradigm(s)	Common task(s)	Interpretive note
Linear / logistic models	Supervised	Regression; classification	Strong baseline; interpretable coefficients under assumptions; supports explicit regularization.
Decision trees	Supervised	Classification; regression	Nonlinear partitions; interpretable small trees; unstable without ensembles.
Random forests	Supervised	Classification; regression	Bagged randomized trees; robust nonlinear baseline; variable importance needs careful interpretation.
k-nearest neighbours	Supervised (typical)	Classification; regression	Instance-based; sensitive to scale, distance metric and dimensionality.
Support-vector methods	Supervised / one-class	Classification; regression; anomaly detection	Kernelized margins; one-class variant estimates support of a distribution.
Gradient boosting	Supervised	Classification; regression; ranking	Sequential additive fitting to residual/gradient signal; flexible but tuning-sensitive.
Neural networks / deep learning	Supervised; self-supervised; semi-supervised; RL	Classification; regression; representation; generation; anomaly detection; control	Highly flexible function class; architecture and objective determine task.
K-means / DBSCAN	Unsupervised	Clustering	Centroid-based versus density-based cluster structure.
Apriori / FP-growth	Unsupervised pattern mining	Association / frequent-pattern mining	Discover frequent itemsets and rules; not anomaly-detection methods.
PCA / NMF / manifold methods	Unsupervised	Dimensionality reduction / representation	Different objectives imply different geometry and interpretation.
Q-learning / policy gradients	Reinforcement learning	Sequential control / decision making	Learn value functions or policies from reward feedback.

6. Model Development, Leakage Control and Evaluation

Taxonomy alone does not guarantee a valid ML study. Figure 2 presents a leakage-aware workflow in which the split between model-development data and final evaluation data occurs before any transformation that can learn from observations. Data leakage arises when information that would not be available at the intended prediction time, or information from validation/test observations, influences feature construction, preprocessing, selection, tuning, or training. Leakage can produce striking but non-reproducible performance. Kapoor and Narayanan (2023) documented leakage-related reproducibility problems across numerous ML-based scientific studies, making leakage control a methodological requirement rather than a minor implementation detail.

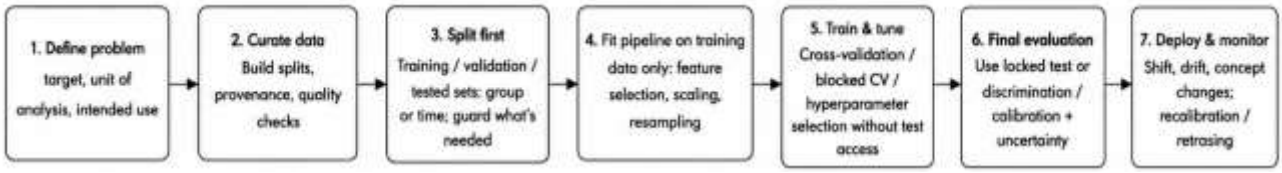

Figure 2. Leakage-aware machine-learning development and evaluation workflow. *Note.* All transformations that learn from data belong inside the model-development pipeline and must be fitted without access to the locked test data. Adapted conceptually from validation and leakage principles discussed by Cawley and Talbot (2010), Varma and Simon (2006), and Kapoor and Narayanan (2023).

Preprocessing should be treated as part of the fitted model pipeline. Imputation, scaling, feature selection, dimensionality reduction, oversampling, target encoding, and other learned transformations must be estimated using training folds only and then applied to held-out folds. Hyperparameters should be selected using a validation set or cross-validation confined to model-development data. When extensive model or hyperparameter selection is performed, nested cross-validation can reduce the optimism that results from using the same resampling process both to choose the model and to estimate its performance (Cawley & Talbot, 2010; Varma & Simon, 2006). The final locked test set should be used only after the pipeline and decision rules are fixed.

Evaluation metrics must match the target and operational costs. For classification, accuracy can be misleading when classes are highly imbalanced. ROC curves summarize the trade-off between sensitivity and false-positive rate across thresholds (Fawcett, 2006), while precision-recall curves can be more informative when the positive class is rare (Saito & Rehmsmeier, 2015). Threshold metrics such as sensitivity, specificity, precision, and F1 should be reported with the class definition and threshold-selection procedure. The choice of operating point should be based on the intended use, not selected post hoc to maximize test-set performance.

Discrimination is not the same as calibration. A model can rank higher-risk observations correctly while assigning probabilities that are systematically too high or too low. The Brier score provides a proper scoring rule for probabilistic predictions (Brier, 1950), and reliability diagrams or calibration curves show agreement between predicted and observed probabilities. Niculescu-Mizil and Caruana (2005) demonstrated that supervised learning methods differ in probability calibration, and Guo et al. (2017) showed that modern neural networks can be poorly calibrated even when classification accuracy is high. When predicted probabilities inform decisions, calibration assessment is therefore essential.

For regression, error should be summarized with measures that reflect the intended penalty for deviations, commonly mean absolute error (MAE), mean squared error (MSE) or root mean squared error (RMSE), together with uncertainty where appropriate. For clustering, no single internal score establishes substantive validity; the silhouette is one widely used measure of cohesion and separation (Rousseeuw, 1987), but domain interpretation and stability remain important. For anomaly detection, performance is especially sensitive to prevalence, alert thresholds, and the cost of false alarms, so precision-recall measures and operational alert rates are often more informative than accuracy.

Reproducibility also requires reporting enough of the pipeline to permit independent reconstruction: data provenance and eligibility, train/validation/test definitions, random seeds where material, preprocessing, feature selection, hyperparameter search space, model-selection criterion, final thresholds, software versions, and uncertainty around performance. Pedregosa et al. (2011) illustrated the value of consistent, composable APIs for ML workflows, while Sculley et al. (2015) emphasized that production ML systems accumulate dependencies and technical debt beyond the model code itself.

7. Application Areas and Published Empirical Evidence

Machine learning has been applied to vision, medicine, natural language processing, recommender systems, cybersecurity, manufacturing, and scientific discovery. Representative application claims are therefore linked to primary empirical studies. Table 4 reports selected landmark demonstrations together with each study's sample or evaluation scale and its reported result.

In computer vision, Krizhevsky et al. (2012) trained a deep convolutional neural network on approximately 1.2 million ILSVRC-2010 training images across 1,000 classes and reported test top-1 and top-5 error rates of 37.5% and 17.0%, respectively. A variant entered in ILSVRC-2012 achieved a top-5 test error of 15.3%, compared with 26.2% for the second-best entry. ImageNet itself provided a large hierarchical image database that helped make large-scale visual recognition benchmarks possible (Deng et al., 2009), while residual networks later made substantially deeper architectures easier to optimize (He et al., 2016). These studies illustrate a general pattern: advances often reflect the combined effect of data scale, architectures, optimization, and computation rather than an algorithm name in isolation.

Medical imaging provides both evidence of potential and a warning against casual generalization. Gulshan et al. (2016) developed a retinal-image classifier using 128,175 images and evaluated it on two validation sets containing 9,963 and 1,748 images. Among the 8,788 and 1,745 fully gradable images used for the referable-diabetic-retinopathy analysis, the predefined high-specificity operating point yielded sensitivity/specificity of 90.3%/98.1% and 87.0%/98.5%, respectively. Esteva et al. (2017) trained on 129,450 clinical images representing 2,032 diseases and compared two binary skin-lesion tasks against 21 board-certified dermatologists, reporting performance on par with the tested experts. De Fauw et al. (2018) trained a deep system on 14,884 optical-coherence-tomography scans and reported referral performance that reached or exceeded expert performance on the study task. These demonstrations remain specific to their data, reference standards, operating thresholds, and validation settings.

In sequential decision making, Mnih et al. (2015) evaluated a deep Q-network across 49 Atari 2600 games using pixels and game score as inputs. It outperformed the leading reinforcement-learning comparators on 43 games and achieved more than 75% of a professional human tester's normalized score on 29 games. Silver et al. (2016) combined policy and value networks with tree search; AlphaGo won 494 of 495 games (99.8%) against other Go programs and won all five formal games against European champion Fan Hui. In scientific discovery, AlphaFold's CASP14 evaluation represented a major advance in protein-structure prediction: Jumper et al. (2021) reported a median backbone accuracy of 0.96 Å r.m.s.d.95 on 87 domains, compared with 2.8 Å for the next-best method. These results show that ML can be embedded in very different problem structures, from perception to control to scientific inference.

Other domains are supported by equally specific methodological literatures. Matrix-factorization methods underpin many recommender systems (Koren et al., 2009). Machine learning has been studied for predictive maintenance in industrial equipment (Susto et al., 2015) and for network intrusion detection, although Sommer and Paxson (2010) cautioned that operational network environments violate many closed-world assumptions common in benchmark ML studies. Electronic health record prediction at scale likewise illustrates both opportunity and the importance of careful validation (Rajkomar et al., 2018).

Table 4 provides the quantitative evidence underlying several major application claims. The examples are illustrative, not a representative sample of application-wide effectiveness, and the results were neither recalculated nor pooled.

Table 4. Selected landmark primary studies and reported empirical results

Reference	Task	Dataset / evaluation scale	Reported result	Interpretation constraint
Krizhevsky et al. (2012)	Image classification	Approximately 1.2 million ILSVRC-2010 training images; 1,000 classes; separate ILSVRC-2012 entry	ILSVRC-2010 test errors: 37.5% top-1 and 17.0% top-5; ILSVRC-2012 top-5 test error: 15.3%.	Large-scale benchmark result; not a statement of performance on arbitrary image domains.
Gulshan et al. (2016)	Referable diabetic retinopathy detection	128,175 development images; validation sets of 9,963 and 1,748 images (8,788 and 1,745 fully gradable for the primary analysis)	High-specificity operating point: 90.3% sensitivity / 98.1% specificity and 87.0% / 98.5% on the two validation sets.	Study used ophthalmologist-derived reference standards and preselected operating points.
Esteva et al. (2017)	Skin-lesion classification	129,450 clinical images; 2,032 diseases; comparison with 21 board-certified dermatologists	CNN performance was reported as on par with tested experts on two binary tasks.	Expert comparison was task- and dataset-specific; not a general replacement claim.
Mnih et al. (2015)	Deep reinforcement learning	49 Atari 2600 games	DQN outperformed leading reinforcement-learning comparators on 43 games and exceeded 75% of a professional human tester's normalized score on 29 games.	Game environment provides a controlled reward signal unlike many real-world deployments.
Silver et al. (2016)	Computer Go	Tournament evaluation against Go programs and Fan Hui	494 wins in 495 games against other Go programs (99.8%); 5-0 in formal games against European champion Fan Hui.	Combined supervised learning, reinforcement learning and tree search.
De Fauw et al. (2018)	OCT retinal diagnosis/referral	14,884 OCT scans used for training	Referral performance reached or exceeded experts on the evaluated referral task.	Two-stage architecture separated tissue segmentation from clinical decision network.
Jumper et al. (2021)	Protein-structure prediction	CASP14 evaluation; 87 domains reported for accuracy comparison	Median backbone accuracy 0.96 Å r.m.s.d.95 vs 2.8 Å for the next-best method.	A benchmark/scientific evaluation; downstream biological interpretation still requires domain validation.

8. Challenges, Limitations and Risk Controls

The limitations of ML are better framed as identifiable failure modes than as blanket statements that algorithms are inaccurate or lack intelligence. Table 5 groups major risks with evidence and practical controls. The first category is data validity. Models can only learn relationships present in the training information, and those relationships may reflect measurement error, selection bias, missingness, historical

inequity, or proxy variables. Increasing sample size does not remove systematic error in the data-generating process (Domingos, 2012; Obermeyer et al., 2019).

Fairness failures can be substantial and measurable. Buolamwini and Gebru (2018) evaluated three commercial gender-classification systems and found much higher error rates for darker-skinned women than for lighter-skinned men, with error rates reaching 34.7% for the former group compared with a maximum of 0.8% for the latter across the evaluated systems. Obermeyer et al. (2019) showed how a health-management algorithm's use of cost as a proxy for need produced racial bias. These examples demonstrate why aggregate accuracy is insufficient: subgroup performance, label construction, proxy choice, and the social context of data collection must be examined.

Interpretability is another deployment concern, especially when model outputs affect high-stakes decisions. LIME explains an individual prediction by fitting a sparse local surrogate around the observation (Ribeiro et al., 2016), while SHAP links feature attribution to Shapley-value principles and unifies a class of additive explanation methods (Lundberg & Lee, 2017). Such tools can illuminate model behaviour, but an explanation of what a model used is not automatically a causal explanation of the real-world phenomenon. Explanations should therefore be validated for stability, fidelity, and usefulness to the intended decision-maker.

Privacy and security risks arise because trained models and data pipelines can expose information or be manipulated. Membership-inference attacks can reveal whether a record was part of a model's training set (Shokri et al., 2017). Differentially private learning provides a formal mechanism for limiting the influence of individual training examples, though often with a privacy-utility trade-off (Abadi et al., 2016). Adversarial examples show that small, strategically chosen input perturbations can cause neural networks to make incorrect predictions (Goodfellow et al., 2015). Secure deployment therefore requires threat modelling, access controls, privacy-preserving procedures where appropriate, and robustness testing rather than assuming that predictive accuracy implies security.

A further risk is hidden technical debt. Production ML depends on data sources, feature pipelines, configuration, monitoring, retraining schedules, downstream consumers, and changing external environments. Sculley et al. (2015) described how entanglement and feedback loops can make ML systems costly to maintain even when the model itself is compact. Monitoring must therefore include not only model accuracy but input drift, missingness, calibration, subgroup performance, operational latency, and changes in decision consequences.

Claims about employment should also be evidence-based. Automation can substitute for some tasks while complementing others, and its effects differ across occupations, locations, and time. Autor (2015) emphasized the continuing reorganization of tasks around technology, whereas Acemoglu and Restrepo (2020) found negative local labour-market effects associated with industrial robot exposure in the United States. It is therefore too strong to state that ML necessarily 'eliminates decision-making jobs'; a more defensible conclusion is that ML changes task composition, skill requirements, and organizational workflows, with distributional effects that require empirical study.

Finally, ML performance is bounded by the problem formulation and deployment distribution. Strong benchmark performance does not establish human-like common sense, causal understanding, or validity under arbitrary distribution shift. The appropriate scientific claim is narrower: a model estimates a function optimized under specified data, objectives, and assumptions. This framing avoids anthropomorphism while still recognizing the remarkable empirical capabilities demonstrated in vision, language, control, and scientific prediction.

Table 5 links the principal risks to concrete evidence and controls. This framing is more actionable than an undifferentiated list of disadvantages because it identifies where in the ML lifecycle a failure can arise and how it can be investigated.

Table 5. Major machine-learning risks, supporting evidence and practical controls

Risk area	How the failure arises	References	Practical controls
Data leakage / selection bias	Information from test data or future outcomes enters model development; performance estimate becomes optimistic.	Cawley & Talbot (2010); Varma & Simon (2006); Kapoor & Narayanan (2023)	Split before learned preprocessing; pipelines; nested CV when selecting models; locked test set; temporal/group split when required.
Bias and unfairness	Aggregate performance hides subgroup disparities; labels/proxies encode historical inequity.	Buolamwini & Gebru (2018); Obermeyer et al. (2019)	Audit labels/proxies; report subgroup metrics; assess representativeness; conduct fairness and impact review.
Poor calibration	Predicted probabilities do not match observed event rates.	Niculescu-Mizil & Caruana (2005); Guo et al. (2017)	Calibration curves/Brier score; recalibration on appropriate validation data; threshold analysis.
Opacity / interpretability	Stakeholders cannot readily determine why a complex model produced an output.	Ribeiro et al. (2016); Lundberg & Lee (2017)	Use interpretable baselines; local/global explanations; validate explanation fidelity and stability; document limitations.
Privacy leakage	Model or pipeline reveals sensitive training information.	Shokri et al. (2017); Abadi et al. (2016)	Data minimization, access control, differential privacy where suitable, privacy testing and governance.
Adversarial / security failure	Inputs are intentionally manipulated to induce errors.	Goodfellow et al. (2015)	Threat modelling, adversarial/robustness testing, secure interfaces, monitoring and incident response.
Technical debt / drift	Data dependencies, feedback loops and changing environments degrade system reliability.	Sculley et al. (2015)	Version data/models; monitor drift and calibration; define retraining/rollback; maintain data contracts and tests.
Workforce effects	Automation changes tasks, demand for skills and distribution of work.	Autor (2015); Acemoglu & Restrepo (2020)	Evaluate task-level effects; reskilling; human oversight; organizational redesign rather than deterministic job-loss claims.

Note. Controls are general methodological safeguards; domain-specific regulation, governance, and safety requirements may impose additional obligations.

9. Discussion

The central finding of this review is conceptual rather than numerical: machine-learning systems are most clearly classified using at least two independent axes. The first asks how the learning signal is obtained; the second asks what task or output is sought. A third descriptive layer then identifies the algorithmic family and model architecture. Cross-cutting procedures—regularization, feature engineering, cross-validation, calibration, privacy controls, and deployment monitoring—should not be forced into either of the first two axes.

The two-axis view also resolves common inconsistencies in machine-learning terminology. Logistic regression belongs under classification when the target is categorical; association-rule mining is represented by methods such as Apriori and FP-growth rather than anomaly-detection methods; neural networks are a broad model family rather than a clustering algorithm per se; regularization is a complexity-control technique rather than a functional task; and supervised learning is designed to infer outputs for new observations whose true labels are unknown at prediction time. These distinctions directly affect how researchers select and describe methods.

The review also shows why a taxonomy should remain flexible. Contemporary systems often combine paradigms: a representation may be learned self-supervised on a large unlabelled corpus, fine-tuned with supervised labels, calibrated on a validation set, and embedded in a downstream decision system. AlphaGo combined supervised learning from expert games with reinforcement learning and tree search (Silver et al., 2016). BERT used self-supervised pretraining followed by supervised downstream adaptation (Devlin et al., 2019). The taxonomy is therefore not intended to place every system in exactly one box; its value is to label each design dimension accurately.

A second implication concerns evaluation. High performance is meaningful only when the evaluation design prevents information leakage and matches the intended use. Discrimination metrics, calibration, subgroup performance, uncertainty, and operational thresholds answer different questions. A model selected because it has the highest cross-validated AUC may still be poorly calibrated, unfair across subgroups, unstable under distribution change, or impractical to deploy. Model comparison should

therefore include strong statistical baselines and should not assume that increased complexity guarantees better generalization (Breiman, 2001b; Domingos, 2012; Guo et al., 2017).

This review has limitations. As a critical narrative synthesis, it does not provide exhaustive database coverage, a protocol-registered search, formal risk-of-bias scoring, or meta-analytic effect estimates. The purposively selected empirical examples illustrate distinct tasks and evaluation designs but do not constitute a representative sample of performance across application domains. The proposed taxonomy is a conceptual synthesis and has not been tested through independent coding or inter-rater agreement. Its application can nevertheless be made auditable by reporting each system's learning signal, task, model family, data partitions, evaluation design, and deployment context. Future work should test classification reliability and extend the framework to rapidly evolving areas, including foundation models, multimodal learning, causal ML, federated learning, and AI safety.

10. Conclusion

Machine learning is not a single algorithmic technique but a family of learning systems whose design can be described by the source of the learning signal, the task being solved, the algorithm or architecture used, and the procedures used to control and evaluate the model. Separating these dimensions eliminates common category errors and provides a more durable framework for understanding a field in which the same algorithm can solve multiple tasks and the same task can be approached by multiple learning paradigms.

A technically sound classification therefore distinguishes supervised, unsupervised, semi-supervised, self-supervised, and reinforcement learning from tasks such as classification, regression, clustering, anomaly detection, association-rule mining, dimensionality reduction, and generative modelling. Regularization, validation, calibration, privacy protection, explainability, and deployment monitoring are cross-cutting controls rather than parallel task categories. The literature further shows that credible ML claims depend on rigorous data partitioning, leakage control, appropriate metrics, calibration, subgroup evaluation, and clear reporting of the intended use.

Future educational and review treatments of ML should preserve these distinctions and anchor claims in original literature. As the field evolves, taxonomies should be extended by adding new paradigms or tasks without collapsing conceptually different levels. This approach supports clearer teaching, more defensible method selection, and more reproducible application of machine learning across scientific and industrial domains.

Acknowledgements

This study forms part of the PhD dissertation of the lead author University of Port Harcourt, Nigeria under the supervision and scholarly guidance of the other authors.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 308–318). Association for Computing Machinery. https://doi.org/10.1145/2976749.2978318

Acemoglu, D., & Restrepo, P. (2020). Robots and jobs: Evidence from US labor markets. Journal of Political Economy, 128(6), 2188–2244. https://doi.org/10.1086/705716

Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data (pp. 207–216). Association for Computing Machinery. https://doi.org/10.1145/170035.170072

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In Proceedings of the 20th International Conference on Very Large Data Bases (pp. 487–499). Morgan Kaufmann.

Autor, D. H. (2015). Why are there still so many jobs? The history and future of workplace automation. Journal of Economic Perspectives, 29(3), 3–30. https://doi.org/10.1257/jep.29.3.3

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In Proceedings of the Eleventh Annual Conference on Computational Learning Theory (pp. 92–100). Association for Computing Machinery. https://doi.org/10.1145/279943.279962

Breiman, L. (2001a). Random forests. Machine Learning, 45, 5–32. https://doi.org/10.1023/A:1010933404324

Breiman, L. (2001b). Statistical modeling: The two cultures (with comments and a rejoinder by the author). Statistical Science, 16(3), 199–231. https://doi.org/10.1214/ss/1009213726

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). LOF: Identifying density-based local outliers. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (pp. 93–104). Association for Computing Machinery. https://doi.org/10.1145/335191.335388

Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. Monthly Weather Review, 78(1), 1–3. https://doi.org/10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2

Buolamwini, J., & Gebru, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. Proceedings of Machine Learning Research, 81, 77–91.

Cawley, G. C., & Talbot, N. L. C. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. Journal of Machine Learning Research, 11, 2079–2107.

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. Proceedings of Machine Learning Research, 119, 1597–1607.

Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. Journal of the American Statistical Association, 74(368), 829–836. https://doi.org/10.1080/01621459.1979.10481038

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20, 273–297. https://doi.org/10.1007/BF00994018

Cover, T. M., & Hart, P. E. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21–27. https://doi.org/10.1109/TIT.1967.1053964

De Fauw, J., Ledsam, J. R., Romera-Paredes, B., Nikolov, S., Tomasev, N., Blackwell, S., Askham, H., Glorot, X., O'Donoghue, B., Visentin, D., van den Driessche, G., Lakshminarayanan, B., Meyer, C., Mackinder, F., Bouton, S., Ayoub, K., Chopra, R., King, D., Karthikesalingam, A., ... Ronneberger, O. (2018). Clinically applicable deep learning for diagnosis and referral in retinal disease. Nature Medicine, 24, 1342–1350. https://doi.org/10.1038/s41591-018-0107-6

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition (pp. 248–255). IEEE. https://doi.org/10.1109/CVPR.2009.5206848

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of NAACL-HLT 2019, Volume 1 (pp. 4171–4186). Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423

Domingos, P. (2012). A few useful things to know about machine learning. Communications of the ACM, 55(10), 78–87. https://doi.org/10.1145/2347736.2347755

Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (2004). Least angle regression. The Annals of Statistics, 32(2), 407–499. https://doi.org/10.1214/009053604000000067

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (pp. 226–231). AAAI Press.

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. Nature, 542, 115–118. https://doi.org/10.1038/nature21056

Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1), 119–139. https://doi.org/10.1006/jcss.1997.1504

Friedman, J. H. (1991). Multivariate adaptive regression splines. The Annals of Statistics, 19(1), 1–67. https://doi.org/10.1214/aos/1176347963

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. The Annals of Statistics, 29(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In Advances in Neural Information Processing Systems 27 (pp. 2672–2680).

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. In International Conference on Learning Representations.

Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., Cuadros, J., Kim, R., Raman, R., Nelson, P. C., Mega, J. L., & Webster, D. R. (2016). Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. JAMA, 316(22), 2402–2410. https://doi.org/10.1001/jama.2016.17216

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. Proceedings of Machine Learning Research, 70, 1321–1330.

Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (pp. 1–12). Association for Computing Machinery. https://doi.org/10.1145/335191.335372

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (pp. 770–778). IEEE. https://doi.org/10.1109/CVPR.2016.90

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. Science, 313(5786), 504–507. https://doi.org/10.1126/science.1127647

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1), 55–67. https://doi.org/10.1080/00401706.1970.10488634

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. Science, 349(6245), 255–260. https://doi.org/10.1126/science.aaa8415

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., ... Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. Nature, 596, 583–589. https://doi.org/10.1038/s41586-021-03819-2

Kapoor, S., & Narayanan, A. (2023). Leakage and the reproducibility crisis in machine-learning-based science. Patterns, 4(9), 100804. https://doi.org/10.1016/j.patter.2023.100804

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In Proceedings of the 14th International Joint Conference on Artificial Intelligence (Vol. 2, pp. 1137–1143). Morgan Kaufmann.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8), 30–37. https://doi.org/10.1109/MC.2009.263

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems 25 (pp. 1097–1105).

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521, 436–444. https://doi.org/10.1038/nature14539

Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. Nature, 401, 788–791. https://doi.org/10.1038/44565

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. In 2008 Eighth IEEE International Conference on Data Mining (pp. 413–422). IEEE. https://doi.org/10.1109/ICDM.2008.17

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In Advances in Neural Information Processing Systems 30 (pp. 4765–4774).

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability (Vol. 1, pp. 281–297). University of California Press.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H.,

Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. Nature, 518, 529–533. https://doi.org/10.1038/nature14236

Niculescu-Mizil, A., & Caruana, R. (2005). Predicting good probabilities with supervised learning. In Proceedings of the 22nd International Conference on Machine Learning (pp. 625–632). Association for Computing Machinery. https://doi.org/10.1145/1102351.1102430

Obermeyer, Z., Powers, B., Vogeli, C., & Mullainathan, S. (2019). Dissecting racial bias in an algorithm used to manage the health of populations. Science, 366(6464), 447–453. https://doi.org/10.1126/science.aax2342

Pearson, K. (1901). 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. https://doi.org/10.1080/14786440109462720

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

Quinlan, J. R. (1986). Induction of decision trees. Machine Learning, 1, 81–106. https://doi.org/10.1007/BF00116251

Rajkomar, A., Oren, E., Chen, K., Dai, A. M., Hajaj, N., Hardt, M., Liu, P. J., Liu, X., Marcus, J., Sun, M., Sundberg, P., Yee, H., Zhang, K., Zhang, Y., Flores, G., Duggan, G. E., Irvine, J., Le, Q., Litsch, K., ... Dean, J. (2018). Scalable and accurate deep learning with electronic health records. npj Digital Medicine, 1, 18. https://doi.org/10.1038/s41746-018-0029-1

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). 'Why should I trust you?': Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1135–1144). Association for Computing Machinery. https://doi.org/10.1145/2939672.2939778

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 20, 53–65. https://doi.org/10.1016/0377-0427(87)90125-7

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323, 533–536. https://doi.org/10.1038/323533a0

Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLOS ONE, 10(3), e0118432. https://doi.org/10.1371/journal.pone.0118432

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. IBM Journal of Research and Development, 3(3), 210–229. https://doi.org/10.1147/rd.33.0210

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. Neural Computation, 13(7), 1443–1471. https://doi.org/10.1162/089976601750264965

Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., & Dennison, D. (2015). Hidden technical debt in machine learning systems. In Advances in Neural Information Processing Systems 28 (pp. 2503–2511).

Shmueli, G. (2010). To explain or to predict? Statistical Science, 25(3), 289–310. https://doi.org/10.1214/10-STS330

Shokri, R., Stronati, M., Song, C., & Shmatikov, V. (2017). Membership inference attacks against machine learning models. In 2017 IEEE Symposium on Security and Privacy (pp. 3–18). IEEE. https://doi.org/10.1109/SP.2017.41

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. Nature, 529, 484–489. https://doi.org/10.1038/nature16961

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., Cubuk, E. D., Kurakin, A., & Li, C.-L. (2020). FixMatch: Simplifying semi-supervised learning with consistency and confidence. In Advances in Neural Information Processing Systems 33 (pp. 596–608).

Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. In 2010 IEEE Symposium on Security and Privacy (pp. 305–316). IEEE. https://doi.org/10.1109/SP.2010.25

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15, 1929–1958.

Susto, G. A., Schirru, A., Pampuri, S., McLoone, S., & Beghi, A. (2015). Machine learning for predictive maintenance: A multiple classifier approach. IEEE Transactions on Industrial Informatics, 11(3), 812–820. https://doi.org/10.1109/TII.2014.2349359

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. Machine Learning, 3, 9–44. https://doi.org/10.1007/BF00115009

Tenenbaum, J. B., de Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. Science, 290(5500), 2319–2323. https://doi.org/10.1126/science.290.5500.2319

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

Turing, A. M. (1950). Computing machinery and intelligence. Mind, 59(236), 433–460. https://doi.org/10.1093/mind/LIX.236.433

Valiant, L. G. (1984). A theory of the learnable. Communications of the ACM, 27(11), 1134–1142. https://doi.org/10.1145/1968.1972

van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9, 2579–2605.

Varma, S., & Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. BMC Bioinformatics, 7, 91. https://doi.org/10.1186/1471-2105-7-91

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems 30 (pp. 5998–6008).

Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. Machine Learning, 8, 279–292. https://doi.org/10.1007/BF00992698

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 8, 229–256. https://doi.org/10.1007/BF00992696

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67(2), 301–320. https://doi.org/10.1111/j.1467-9868.2005.00503.x