

Introduction to Artificial Intelligence and Machine Learning for Pathology

James H. Harrison Jr, MD, PhD; John R. Gilbertson, MD; Matthew G. Hanna, MD; Niels H. Olson, MD; Jansen N. Seheult, MB, BCh, BAO, MSc, MD; James M. Sorace, MD, MS; Michelle N. Stram, MD, MSc

• **Context.**—Recent developments in machine learning have stimulated intense interest in software that may augment or replace human experts. Machine learning may impact pathology practice by offering new capabilities in analysis, interpretation, and outcomes prediction using images and other data. The principles of operation and management of machine learning systems are unfamiliar to pathologists, who anticipate a need for additional education to be effective as expert users and managers of the new tools.

Objective.—To provide a background on machine learning for practicing pathologists, including an overview of algorithms, model development, and performance evaluation; to examine the current status of machine learning in pathology and consider possible roles and requirements for pathologists in local deployment and management of machine learning systems; and to highlight existing challenges and gaps in deployment methodology and regulation.

Data Sources.—Sources include the biomedical and

engineering literature, white papers from professional organizations, government reports, electronic resources, and authors' experience in machine learning. References were chosen when possible for accessibility to practicing pathologists without specialized training in mathematics, statistics, or software development.

Conclusions.—Machine learning offers an array of techniques that in recent published results show substantial promise. Data suggest that human experts working with machine learning tools outperform humans or machines separately, but the optimal form for this combination in pathology has not been established. Significant questions related to the generalizability of machine learning systems, local site verification, and performance monitoring remain to be resolved before a consensus on best practices and a regulatory environment can be established.

(*Arch Pathol Lab Med.* 2021;145:1228–1254; doi: 10.5858/arpa.2020-0541-CP)

Artificial intelligence (AI) is a focus of intense scientific, business, and governmental interest: global AI sci-

tific publications have increased more than 6-fold in the past 20 years to more than 60 000 annually,¹ at least 26 governments have established national AI strategies during the past 4 years,² and daily articles in the business and lay press discuss the future applications and impact of AI. Industry projections variously estimate the global AI market to exceed \$100 billion to \$300 billion by 2025, with an annual growth of 40% to 50%. Health care is an important domain for AI growth. AI publications in the biomedical literature have increased more than 8-fold since 2000 (Figure 1), and the total market for AI in health care is expected to grow from \$856 million in 2017 to more than \$20 billion by 2025, including both software and hardware.³ Although AI technologies have not yet had a large impact on clinical practice, the introduction of software and devices with significant new capabilities could disrupt existing workflows, practice patterns, and reimbursement policies in unpredictable ways. A number of reports and white papers have addressed potential benefits and disruptions in health care from AI, with a generally positive perspective.^{4–6} However, news from initial AI implementations has not been uniformly good: a 2017 industry survey indicated that half of early adopters encountered immature AI products that did not yield anticipated benefits,⁷ a situation perhaps best exemplified by the failure of IBM's Watson Oncology at the MD Anderson Cancer Center (Houston, Texas).⁸

Accepted for publication November 10, 2020.

Published online January 25, 2021.

Supplemental digital content is available for this article at <https://meridian.allenpress.com/aplm> in the October 2021 table of contents.

From the Department of Pathology, University of Virginia School of Medicine, Charlottesville (Harrison); the Departments of Biomedical Informatics and Pathology, University of Pittsburgh, Pittsburgh, Pennsylvania (Gilbertson); the Department of Pathology, Memorial Sloan Kettering Cancer Center, New York, New York (Hanna); the Defense Innovation Unit, Mountain View, California (Olson); and the Department of Pathology, Uniformed Services University, Bethesda, Maryland (Olson); the Department of Pathology, University of Pittsburgh, and Vitalant Specialty Labs, Pittsburgh, Pennsylvania (Seheult); the US Department of Health and Human Services, retired, Lutherville, Maryland (Sorace); and the Department of Forensic Medicine, New York University, and Office of Chief Medical Examiner, New York, New York (Stram).

The authors have no relevant financial interest in the products or companies described in this article.

All authors are members of the Machine Learning Workgroup, College of American Pathologists Informatics Committee.

Corresponding author: James H. Harrison Jr, MD, PhD, Department of Pathology, University of Virginia School of Medicine, PO Box 800168, Charlottesville, VA 22908-0168 (email: james.harrison@virginia.edu).

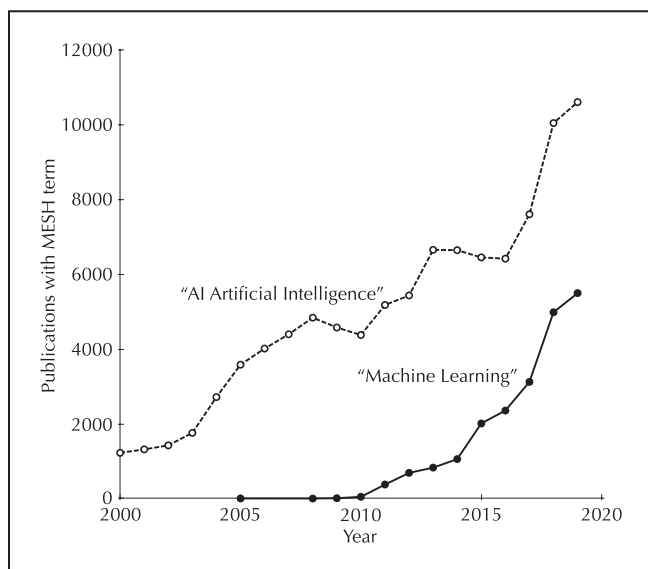


Figure 1. Artificial intelligence and machine learning publications in the biomedical literature. The annual number of publications in PubMed tagged with primary or secondary Medical Subject Headings (MESH) of "AI Artificial Intelligence" (dotted line) or "Machine Learning" (solid line) is shown. The categories are not mutually exclusive. Tagging with "AI Artificial Intelligence" increased more than 8-fold, from 1224 in 2000 to 5941 in 2019. The "Machine Learning" MESH heading was introduced in 2005 and used rarely until after 2010; since then it has increased more than 100-fold, from 49 in 2010 to 5491 in 2019.

As prototype AI products develop, professional organizations and government agencies are attempting to clarify use cases and regulatory strategies for AI in healthcare. The American Medical Association (AMA) has created AI education resources and has released a policy statement with recommendations for the development and deployment of AI systems.⁵ The Canadian Association of Radiologists published a white paper on AI in radiology.⁹ The American College of Radiology has established a Data Science Institute¹⁰ that has created a set of use cases for AI in radiology to guide and promote the development of AI tools for radiology practice. The US Food and Drug Administration (FDA) is collaborating with the American College of Radiology on use case development and is considering methods for evaluating and validating AI software and devices that ensure quality while allowing innovation and optimization.^{11,12} These efforts and the resources they produce are intended to ensure that new products fit health care needs and are implemented in a way that supports care quality.

AI is an umbrella term that covers a variety of techniques for modeling and mimicking human judgment (Figure 2). Current AI systems are termed "narrow" or "weak" AI because they carry out only those tasks for which they were written or trained. Future systems that do not yet exist may implement "general" or "strong" AI that will automatically and independently learn arbitrary tasks. Narrow AI may be implemented in several ways. Expert systems encode human knowledge explicitly as rules in a knowledge base or relationships in an ontology (Figure 2). These rules and relationships are used by a software inference system to derive conclusions about input data. Machine learning systems (Figure 2), in contrast, are not based primarily on human-derived knowledge. Instead, machine learning uses

an algorithm to identify ("learn") repetitive data patterns present in large numbers of example cases and to match new cases to the previously identified patterns.

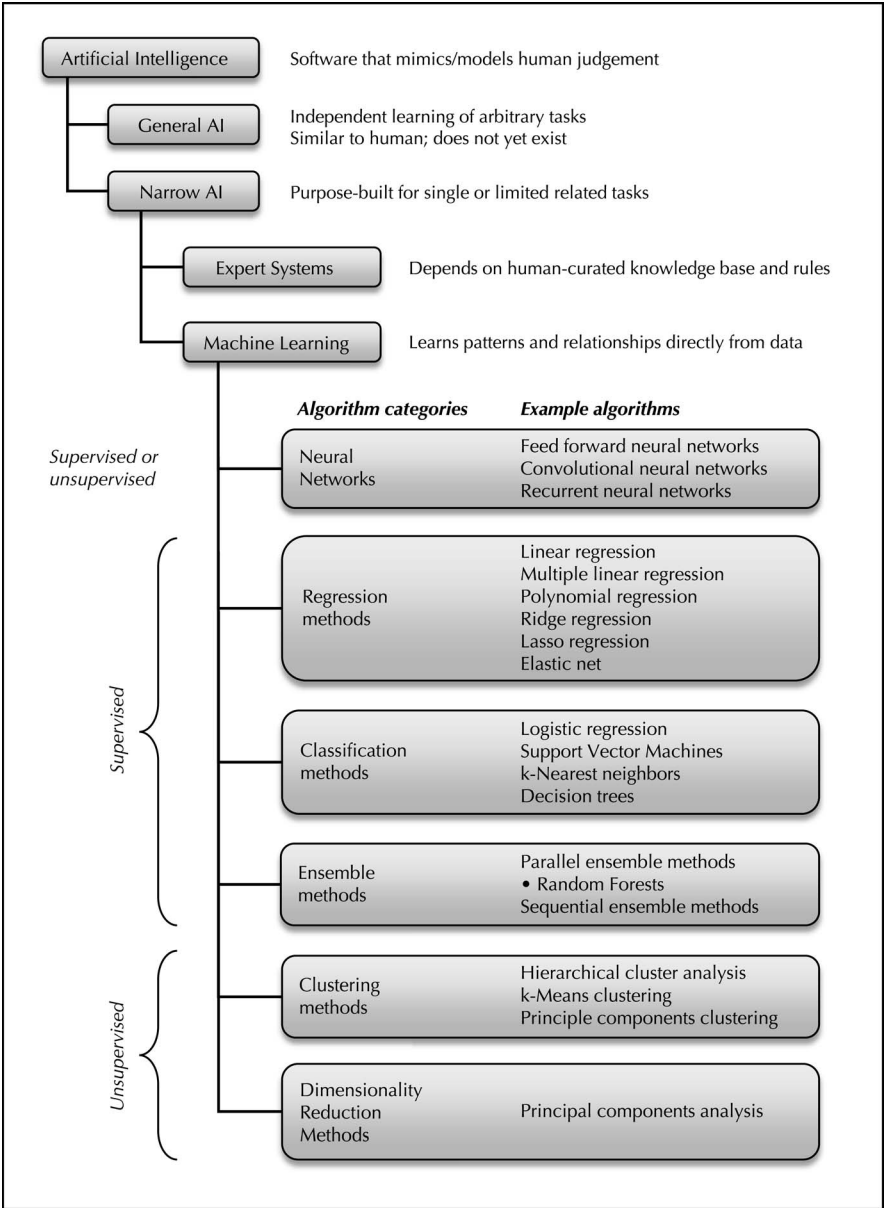
The current excitement about AI applications, including those in medicine, is focused on machine learning.¹³ Although rule-based expert systems have a long history of research and application in medicine,¹⁴ their primary impact has been in systems operating on limited data within specific devices, such as alerting systems in intensive care unit monitors and compound identification software in clinical laboratory mass spectrometry. Modern machine learning systems can handle much more data per case than is practical in rule-based systems (eg, hundreds or thousands of electronic health record data elements or high-resolution medical images) and can identify meaningful patterns in data regardless of whether those patterns are known or their relative importance is well-defined. Hence, machine learning techniques may be effective in complex settings where the development of explicit rules is impractical.

Machine learning has not been deployed widely in clinical medicine because of a variety of limitations. These limitations have been progressively removed during several decades of research and development.¹⁵ Key technical advances include improved training algorithms, development of advanced pattern-recognition algorithms, such as convolutional neural networks for image processing, and high-speed computing hardware, such as graphical processing units. The recent availability of very large training data sets, such as ImageNet (more than 14 million hand-annotated images, <http://www.image-net.org>; accessed October 7, 2020) and text data derived from public online sources and digitized books, has also been crucial in providing adequate volumes of example data for training. These advancements were combined in 2012 to yield breakthrough performance in general image classification¹⁶ and highlight the potential of machine learning.

The new machine learning techniques and data resources stimulated a wave of research and development exploring the use of machine learning applications in a variety of domains, including health care. One of the first machine learning applications to be approved for clinical use interprets funduscopic images automatically for assessment of diabetic retinopathy.^{17,18} Other health care imaging applications in development include classification of gross dermatology images,¹⁹ radiology images,²⁰ and pathology images.²¹ Beyond the imaging domain, machine learning methods have been tested for prediction of clinical course, including applications in sepsis,²² risk assessment in critical care,²³ acute kidney injury,²⁴ and adverse drug events,²⁵ and clinical outcomes, including survival, length of stay,²⁶ and hospital readmission.²⁷ Some researchers are aggressively pursuing application of machine learning to automated diagnosis,²⁸ whereas others see the most appropriate role of machine learning as augmenting clinicians and staff in areas such as patient administration, clinical decision support, patient monitoring, and health care interventions.²⁹

The tension about the optimal relationship between AI applications and health care providers will likely be a central consideration in the design and development of these systems during the next several years, and it may not be resolved until specific applications can be tested in a variety of practice environments. There is early evidence from clinical trials that support of local clinical needs and workflow, and the ability to accommodate local data, are

Figure 2. Types of artificial intelligence (AI) and machine learning. The example machine learning algorithms discussed in this review are listed. Many more algorithms and variations exist.



critical for the successful deployment of AI in health care.³⁰ Furthermore, methods for ensuring the safe and effective operation of AI systems across multiple settings are immature. To reduce risk in the early adoption of AI in health care, the American College of Radiology and the Radiological Society of North America have recommended that the FDA refrain from approving autonomous AI systems until experience is gained with physician-supervised AI, in which physicians could mitigate AI performance problems.³¹ Such a staged approach may allow for the optimization of AI system design in coordination with adaptation of provider roles to fulfill the AMA’s concept of “augmented intelligence,” in which AI tools enhance rather than replace care providers.⁵

APPLICATION OF MACHINE LEARNING TO PATHOLOGY TASKS

Most pathology applications of AI are in relatively early development. There are only a few FDA-approved devices

that use AI, and these support cervical cytology screening and blood/fluid cell classification. In research settings, machine learning has been used successfully to classify and grade lung cancer,^{32,33} predict prognosis in lung³⁴ and brain³⁵ cancer, classify colorectal polyps,³⁶ diagnose and classify lymphoma,³⁷ measure breast tumor proliferation,³⁸ identify metastases in lymph nodes,^{39–41} predict bladder cancer recurrence,⁴² diagnose and grade prostate cancer,^{43,44} and identify tumor stroma.⁴⁵ A detailed review of machine learning applications in pathology is available.⁴⁶ Some of this work has been done as part of machine learning competitions, such as the Camlyeon 2016 Challenge and others^{47,48} that compare the performance of algorithms from many research groups. Future applications may calculate scores such as Ki-67 automatically, identify tumor area, count items on slides (tumor-infiltrating lymphocytes, mitoses, organisms, inclusions), predict cancer genetics and prognosis from histologic sections,^{49,50} identify regions of greatest interest on a slide, and triage cases based on slide

content.^{51,52} Machine learning has been applied to a number of clinical laboratory tasks, including prediction from laboratory results of cardiac risk,⁵³ hepatic disease and anemia,⁵⁴ endocrine diagnoses,⁵⁵ and other conditions.⁵⁶ Machine learning has also been used to predict test results based on existing results of other tests,⁵⁷ potentially reducing the need for testing, and to autovariate gas chromatography–mass spectrometry analyses.⁵⁸ In aggregate these studies in both anatomic pathology and laboratory medicine show promising results, although most are limited to highly focused tasks using high-quality images and data sets that may not fully generalize to routine operational settings.

A number of startup and established companies are attempting to translate the pathology imaging results into commercial products. It is important to recognize that the research prototypes mentioned above each implement a relatively narrow task within the full set of tasks that a pathologist would carry out to evaluate a tissue specimen. Useful products will need broader capabilities, likely through expansion of the initial limited prototypes to complementary aggregates of individual machine learning algorithms working in concert. These algorithm aggregates must be validated against larger and more varied image sets than have been used thus far. At this early stage of development, the optimal design of machine learning tools for pathology and their fit into the pathology workflow has not been defined, and substantial work remains to be done. Studies in breast cancer metastasis detection suggest that machine learning tools may effectively augment pathologist performance in a workflow that remains pathologist-centric.^{40,41} Studies in radiology, which as a domain is somewhat ahead of pathology in its engagement with machine learning, have reached similar conclusions^{59,60} and results in both domains are compatible with the AMA's proposal that machine learning should support and extend physician performance.⁵

Machine learning may yield tools useful for pathology decision support and education. An augmented reality microscope using machine learning⁶¹ that can highlight and label important features in the field of vision in real time could increase pathologist efficiency and also be useful in residency training or continuing medical education. Similarity search across histopathologic image libraries^{62,63} could suggest differential diagnoses, display groups of similar cases, or identify difficult-to-distinguish images for training purposes. Tools that integrate imaging, genomics, biomarkers, and prognosis prediction^{49,50} could support more comprehensive and useful pathology reports and provide a broader view of disease.

A recent survey of 487 pathologists in 59 countries (22% in the United States) showed that about 80% (380 of 473) expected AI tools to be incorporated into the laboratory in the near future. Although there was some concern about job loss, more than 70% of pathologists (347 of 476) expressed interest in or excitement about the potential new capabilities.⁶⁴ Pathologists felt that diagnostic decision-making should remain a predominantly human task but that AI tools could improve the prognostic capability of pathologists while decreasing diagnostic error and improving practice quality and efficiency. These sentiments are generally consistent with the goals for AI recommended by the AMA.⁵ Pathologists believed that there were technical challenges in the implementation of AI tools, including methods for validation and performance monitoring, error

detection, and a requirement for a fully digital pathology workflow, and that there needs to be better definition of medicolegal responsibility for the output of AI systems. They also felt there was a need for training in the effective use of AI, including strategies for AI system deployment and evaluation of system performance, a sentiment shared with radiologists.⁶⁵

Machine learning will have an impact on pathology and health care because it is able to address problems that are difficult to automate in other ways, and evidence suggests performance will be at least adequate. It is difficult to predict how transformative or disruptive this impact will be until tools based on machine learning that are appropriate for clinical application begin to take shape. In the meantime, it is prudent to become familiar with the principles upon which machine learning is based and the likely performance characteristics of machine learning systems. Pathologists will use and manage these systems in the future, and in the shorter term they will collaborate with data scientists and engineers to create systems for testing. This review introduces machine learning concepts, terminology, algorithm examples, and development strategies, describes general strengths and weaknesses of machine learning systems, and discusses issues likely to be important in the validation, deployment, and verification of clinical machine learning tools. We intend the review to be for pathologists without a substantial informatics or engineering background, and our goals are to provide an intuitive understanding of key issues in machine learning, improve the ability to collaborate with machine learning experts and read pathology literature about machine learning, and open a gateway to further study.

HOW DO MACHINES LEARN?

Fundamentally, machine learning systems are capable of modifying and optimizing their output in response to input data. The data processed by these systems are generally in the form of populations of *instances*, which may correspond to patients, images, insurance claims, health care providers, events, or other real world or information objects. The instances in a population share a defined set of associated data elements, or features, whose values vary between instances, for example, a defined set of demographic and laboratory test values for a population of patients. Machine learning systems can process instance populations with several to many thousands of features per instance and associate patterns in the features with an instance label or class, such as a diagnosis or probable outcome, or a numeric result, such as time until readmission or disease-free survival. Depending on the type of problem and the form of machine learning, the size of populations necessary for effective learning can range from a few hundred to millions of instances. There are many different machine learning methods and method variations, but there is also a set of generally applicable core concepts.

There has been controversy over whether particular methods should be considered traditional statistical modeling or machine learning.⁶⁶ Statistical models usually define an explicit mathematical relationship between input and output variables, make assumptions about the characteristics and distributions of the input data, and provide confidence intervals for the output variable. Machine learning methods make fewer assumptions about input variables and can identify empirical relationships and

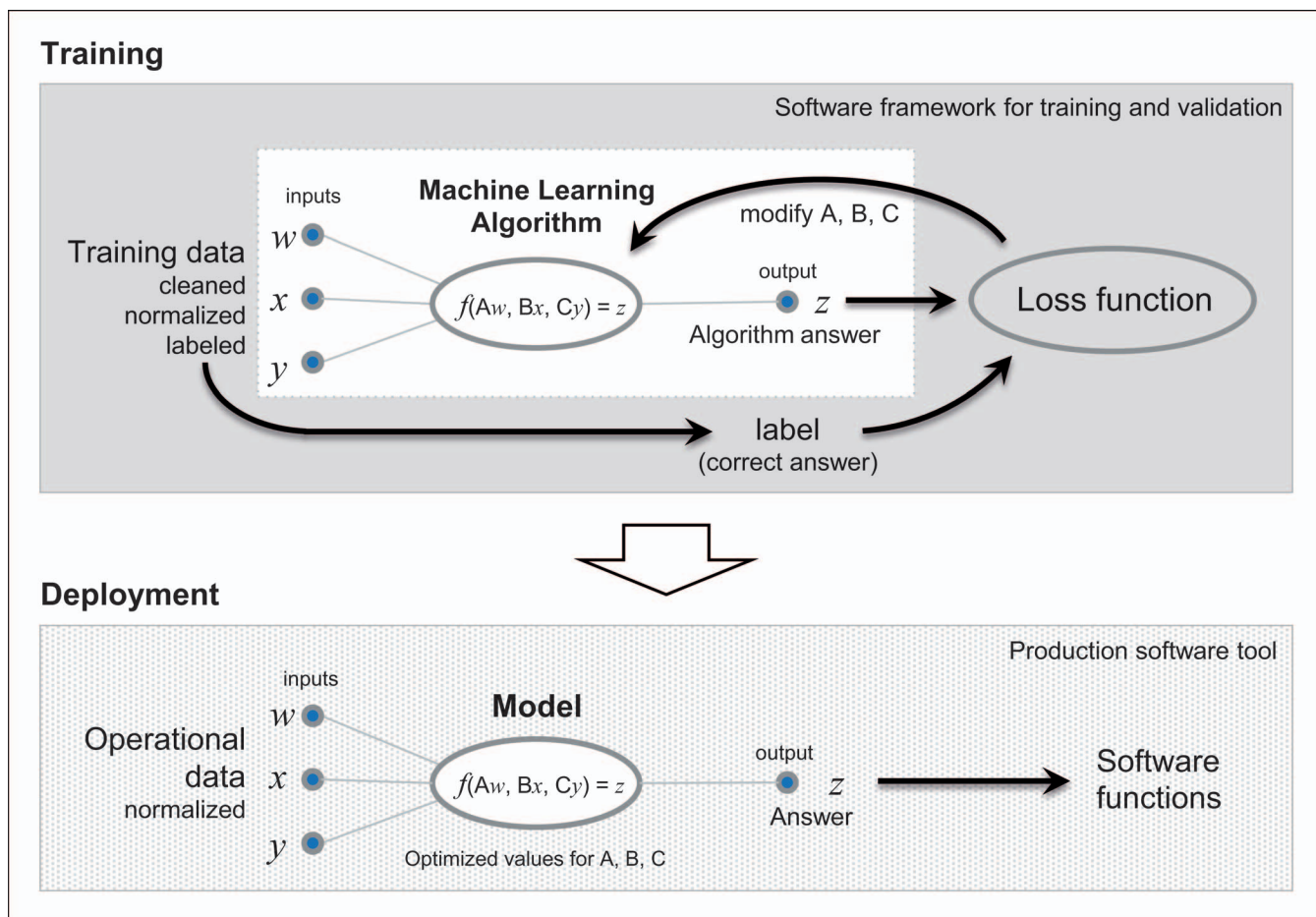


Figure 3. Supervised training of machine learning models. Many machine learning algorithms calculate a numeric or categoric result from input data. When algorithms are trained (Training), correctly labeled instances are fed sequentially to the algorithm. Each instance has characteristic features (w , x , y ; there may be many) that the algorithm uses along with internal weighting and bias factors (A , B , C) to calculate the result (z). During training, the calculated result for each instance (z) is passed to a loss function that compares it to the true value or label. If the calculated result is incorrect, the weights (A , B , C) are adjusted slightly to make the result more correct. This process happens iteratively over many passes through the training data. When a trained algorithm, or model, is deployed (Deployment), it is removed from the training environment and embedded in production software that feeds features (w , x , y) from new, unknown instances into the model and uses the model output (z) in the operation of the software. In static models, the weighting parameters (A , B , C) do not change after the model is removed from the training environment, and the model will perform well as long as the operational data remains consistent with the training data.

associations that characterize groupings of data without necessarily requiring a defined mathematical relationship. In general, machine learning techniques can handle larger numbers of input variables than traditional statistics and may have greater predictive power in complex settings, though without the benefit of a known and explainable mathematical relationship. Both domains seek to fit models that minimize error between predictions and outcomes, and in truth there is a continuum. Some of the methods discussed below are derived from traditional statistics and some developed out of the machine learning domain. The distinction is not particularly important for our purposes.

Most machine learning methods incorporate a central algorithm that processes each instance's features to yield an output for that instance (Figure 3). The output may be a continuous value in *regression* methods, or a category or class label in *classification* methods. The algorithm may implement a simple or complex equation, or it may represent multiple sequential or networked processing steps (as in decision trees or neural networks). During learning,

the output of the algorithm is progressively modified based on the input data. The learning process, referred to as *training*, does not modify the programming code of the algorithm in most forms of machine learning. Instead, the algorithm contains parameters, sometimes called *weights* and *bias values*, that are modified during training to change its output. An algorithm together with its modified parameters is termed a *model*, and the training process is sometimes called building or training a model (see the supplemental digital content at <https://meridian.allenpress.com/aplm> in the October 2021 table of contents for the Machine Learning Glossary containing a list of machine learning terms). A typical model development system contains the algorithm embedded in management software that trains the model by feeding instance data to it, evaluating the accuracy of its output, and modifying its parameters to improve output accuracy (Figure 3, Training).

The training algorithm that evaluates model output and determines how to adjust model parameters is critical and unique to each type of model. Depending on the learning

strategy, the evaluation may determine, for example, whether the output matches a known “true” label for an instance or whether the output of the model improves or degrades the quality of clusters or relationships identified within a population of instances. The parameters of the model are adjusted in relatively large increments when the output deviates significantly from target values, and then in progressively smaller increments as the output becomes closer to optimal. The adjustment process is usually based on minimization of a function, called the objective, cost, or *loss function* depending on the application, that measures the deviation from the correct output. The most common method of optimization, which mathematically descends the curve of the loss function to a minimum value, is called *stochastic gradient descent* and has been shown to be highly effective for most applications.⁶⁷

Trained models incorporate patterns of variation among instance features into their parameter structure such that new instances with feature patterns similar to training instances yield similar output values or categories. If the population used for training is labeled with, for example, clinical outcomes or readmissions, the output can take the form of probabilities or predictions related to those events. Feature patterns producing particular outputs may comprise many features, they may be subtle, and they may not reflect clearly defined mathematical relationships. This ability to use complex, multifactorial data effectively is a strength of machine learning systems. However, it may be difficult to determine independently from training data why an algorithm assigns particular parameter values and weights, and what these values mean, especially for models with many parameters, such as neural networks and ensemble models (see below). Hence, the output of a machine learning algorithm may not be easily explainable, the degree of pattern variation associated with particular outputs may not be straightforward to assess, and model behavior, including error rates, may be difficult to predict other than by exhaustive experimentation. This lack of operational transparency and explainability has been called machine learning’s *black box problem*.

There is a limited subset of machine learning methods, referred to as *instance-based algorithms*,⁶⁸ that do not use a model. Instead, they create a multidimensional map of a population of instances in which the features are the dimensions. A distance or similarity function is used to measure the “closeness” of instances on the map, with similar instances being closer together (ie, clustered) than dissimilar instances. A new instance is mapped into the population data and assigned a category or value based on the identities or average values of its closest neighbors. When optimizing performance, the number of neighbors evaluated when making a class assignment can be varied, and the algorithms may assign weights to neighbors based on distance and category.

In many machine learning projects the goal is to create a static regression or classification model that does not change after training. This approach typically employs a batch learning strategy in which a model is optimized and then removed from the training environment and placed in a software application or device, where it does its work without further modification (Figure 3, Deployment). There are also *incremental learning* systems in which models can continue to adapt to new input while they are being used for routine work. For example, instance-based algorithms (discussed above) can support incremental learning in a straightforward

manner by allowing addition of new, verified instances to the population map. Incremental systems have not been tested widely in medical applications, and because they can alter their performance over time, there are special considerations for their verification and monitoring. Models may be created at the site where they will be deployed, using that site’s data, or they may be created by developers using large aggregates of data from multiple sources. Most medical machine learning applications are likely to use the latter approach, which will require verification against labeled local data as a critical step before deployment.

TYPES OF MACHINE LEARNING

Machine learning algorithms can be broadly classified into 3 categories based on learning strategy: (1) supervised learning, (2) unsupervised learning, and (3) reinforcement learning. *Supervised learning* is essentially learning by example. It depends on a “true” label or value assigned to each instance in the training population, and the learning system seeks to minimize the difference between the output of the model and the “truth” for each instance. For example, a large number of images might be labeled individually by human experts according to their content, and the system would seek to correctly associate patterns of image features with the true label. After training, a good model could process features of new images to yield the correct content labels.

Supervised learning is commonly used for regression and classification problems and is likely to be the basis for the first pathology AI systems that appear. Supervised regression algorithms predict the value of a continuous dependent variable based on instance features that are independent variables,⁶⁷ and are trained against data that are labeled with known dependent values. Supervised classification algorithms process each instance’s features to compute a class assignment or class membership probability, such as a diagnosis, after training against data with known class assignments.⁶⁷ The list of possible classes may be binary (2 classes) or larger. Depending on the data, the class may represent a category to which the instance belongs or a prediction of a future outcome. The application of several supervised methods to pathology tasks has been reviewed recently.⁶⁹

Expert labeling of training data can be demanding, time-consuming, and error prone, particularly when labeling is very detailed, as can be the case with annotation of pathology images. Image annotation may include dividing an image into patches or regions that have structural or clinical meaning (segmentation), and hand labeling these regions using standard terminologies.^{70,71} Techniques are available that may reduce the burden of annotation by automating some or most of the labeling of image features. *Weak supervision* uses a training data set containing a subset of instances with detailed annotation combined with additional instances annotated with less detail or expertise. The combined data set can produce better model performance than the smaller set of detailed annotations alone.^{72,73} *Semisupervised learning* achieves a similar result by training initially with a small annotated data set, using that model to label additional, previously unlabeled instances, and then training again with the combination of the initial and the automatically labeled data.⁷⁴ *Multiple instance learning* avoids detailed annotation of individual instances by automatically learning the important details of

instances grouped under relatively broad labels, such as diagnoses, though this approach typically requires a large number of examples.⁷⁵ Finally, *transfer learning* begins training with a data set from a domain different from the application domain, where more annotated data may be available, and then finishes training with a smaller annotated “tuning” data set from the application domain.^{76,77} Transferrable aspects of the initial training contribute to model performance in the application domain. For example, initial training of an image recognition system on a large set of nonmedical natural images yields transferrable learning of basic image features, such as edges, corners, color gradients, etc. Final training of the system with images from the medical domain defines medical features of interest formed from the previously learned basic image elements and can yield good performance.⁷⁸ Model development strategies that use detailed expert annotation essentially constrain the model to a set of features that are a priori defined as important. Automated feature recognition allows a model to operate in a data-driven style, potentially identifying features from the data different from those felt by experts to be important. These 2 approaches differ in their strengths and weaknesses, and they would be expected to produce models that have different performance and error characteristics.

Unsupervised learning does not use predefined labels or values as outcomes. Instead, the goal is to learn generalizable patterns in the features of a population of instances that most distinctly separate the population into similar subsets.⁶⁷ These patterns may correspond to known categories of instances, but they also may represent previously unrecognized relationships. Discovered patterns can be used to identify associations between features, cluster instances with similar feature profiles, or recognize anomalous instances. New instances may be characterized by their fit into discovered clusters or identified as anomalies. Alternatively, the goals may be to understand the underlying structure and associations in the population data, or to identify data patterns that suggest new hypotheses about the population. In pathology these might take the form, for example, of new functional or structural subgroupings of breast cancer based on subtle histologic variations, shared gene expression profiles, or outcomes.

*Reinforcement learning*⁷⁹ is particularly useful when a system must learn an efficient sequence of actions to reach a goal. It has been used successfully in robotics; game-playing (eg, the recent success of AlphaGo,⁸⁰ developed by a subsidiary of Google); process control, including power grid management; text and speech understanding, and financial trading.⁸¹ This method awards points for progress toward a goal, and it may subtract points for failure to progress. The system seeks to maximize points over time or process steps, rather than match a predefined output or determine structure in data as in supervised and unsupervised learning. Over many iterations of trial and error the system refines action sequences that maximize the reward. Reinforcement learning systems can continue to learn as they operate and are therefore useful in dynamic environments that may change over time.

Within these broad categories of learning strategy, there are many algorithms and variations designed to address particular problems (Figure 2). Some algorithms have broad applicability for both regression and classification in unsupervised and supervised settings, whereas others are more specific. Some have requirements for large amounts of

data for optimal performance, whereas others are more parsimonious, and some algorithms may be more or less suited to different types of data. The following discussion includes a sampling of different types of algorithms used as models in machine learning systems, and many more are available.⁸²

NEURAL NETWORKS

Much of the current excitement about AI in general and in pathology comes from recent breakthroughs in the design and training of neural networks. Some types of neural networks are particularly useful for image classification, and thus of great interest in anatomic pathology. For this reason, we will review neural networks first and then survey selected additional supervised and unsupervised machine learning algorithms.

Neural nets (Figure 4) are networks of (often simple) transfer functions known as “nodes” that each receive inputs related to instance features and produce individual outputs that are aggregated to yield a final output (Figure 4, A). Neural nets are very flexible; depending on their configuration, they can be used in supervised or unsupervised modes for classification or regression. There is a superficial resemblance to networks of biologic neurons, which is the basis for the name. During training, the input weights and internal parameters (eg, bias values) of each transfer function (Figure 4, A) are adjusted independently so that the difference between the actual and optimal overall output is minimized. In essence, training produces multiple simultaneous processing pathways through the network that in sum reflect complex relationships between instance features. Unraveling these processing pathways after training to determine which feature patterns yield particular outputs has been challenging, and this has led to the reputation of neural network models as being unexplainable and opaque “black boxes.”

The recent breakthroughs in AI relate to hardware, methods, and data appropriate for training multilayer neural networks as well as new configurations of multilayer neural network architecture. These kinds of networks contain multiple internal (“hidden”) layers of nodes and are referred to as *deep neural networks*⁸³ (Figure 4, B). *Deep* nets can contain hundreds of layers and many thousands or millions of interconnections, and can represent very complex relationships between large numbers of features. Feed-forward networks have the simplest structure, in which the nodes of each layer are connected only to the nodes of the next layer. Other neural network structures designed for specialized tasks include convolutional and recurrent networks, discussed below. Training these deep networks was not practical previously because of the complexity and volume of adjustments that needed to be made, requiring (1) an efficient processing strategy, (2) high-performance computing, and (3) a very large volume of training data. These problems have been addressed, respectively, with (1) the development of back propagation training methods, which adjust function parameters using loss functions and gradient descent in waves propagating back through the layers from the output to the input of the network, (2) high-speed arrays of graphical processing units, and (3) very large data sets, such as ImageNet, as noted previously. Krizhevsky’s breakthrough work¹⁶ on image classification in 2012 showed that deep neural nets are practical and have the

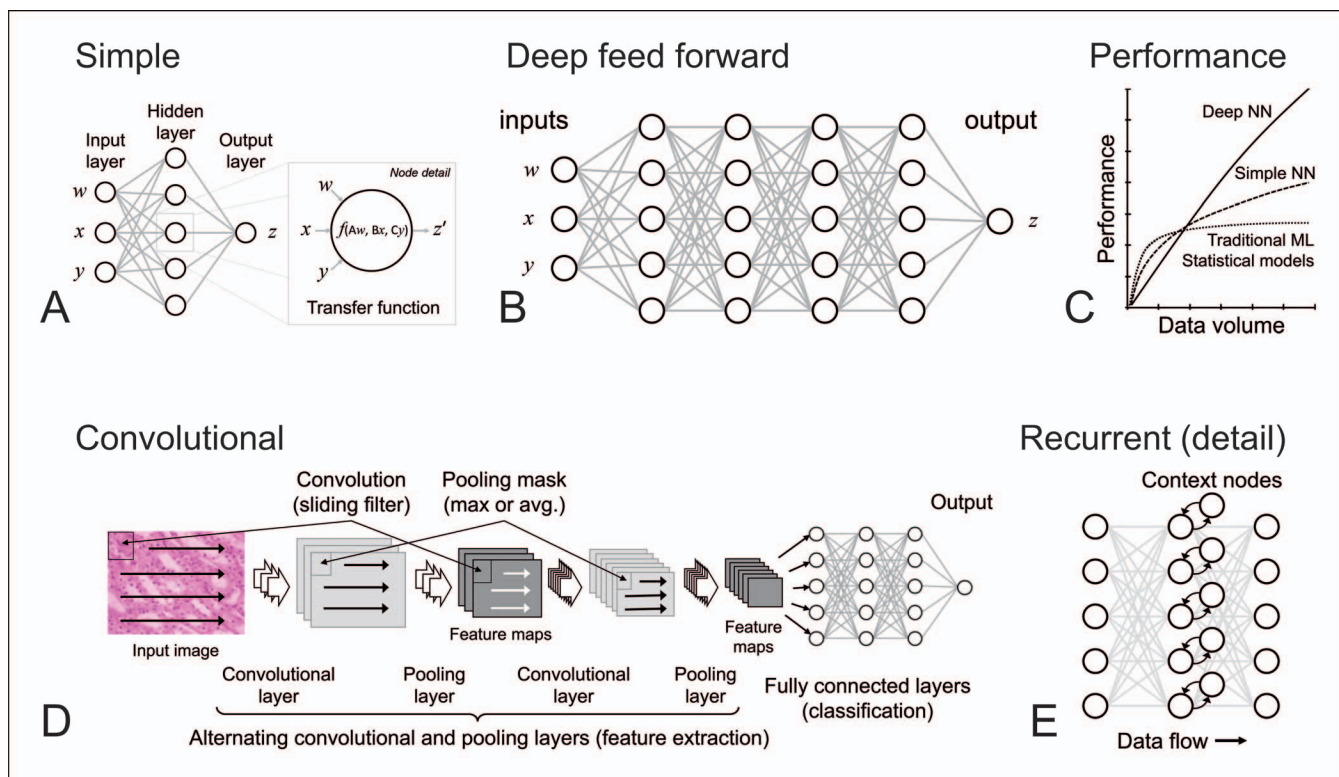


Figure 4. Neural networks. Simple neural networks (A) have a hidden layer of nodes between input and output nodes. Each node implements a transfer function that operates on input data to produce the output. Fully connected feed-forward deep neural networks (B) have multiple hidden layers with all nodes in each layer connected to all nodes in the next downstream layer. Deep neural networks have the potential to perform better than simple neural networks or traditional machine learning and statistical models (C) but require large amounts of training data to do so. Convolutional neural networks (D) process images by extracting features through several initial convolutional and pooling layers that create compact feature maps, before feeding the feature data into a fully connected deep neural net. Recurrent neural networks (E) may have a variety of connections within and between layers. A common pattern is the incorporation of context nodes that store and feed data back to standard nodes, allowing recognition of and response to sequential data patterns.

potential for high performance if large volumes of data are available for training (Figure 4, C).

Convolutional neural networks^{83,84} (CNNs) have a specialized structure for recognizing subpatterns or motifs in data that can be represented as grids or arrays. This capability is most widely used for supervised classification of images based on their content, and is central to many pathology image classification systems.^{32,33,37,39,45} Convolutional neural networks can also be used to classify sequence data (such as time series or gene sequences) based on linear motifs within the overall sequence. The initial layers of a CNN identify characteristic motifs (eg, groupings of pixels) in an image, and then these motifs are used as image features in additional layers with a traditional feed-forward structure that classify the image (Figure 4, D). Feature identification occurs in the initial part of the network using alternating convolutional and pooling layers. The first convolutional layer receives the image data as a 2-dimensional array of pixel brightness values (in a color image there are 3 arrays, referred to as channels representing red, green, and blue values). Small 2-dimensional arrays of weight values called convolutions or filters are applied repeatedly to the image, stepping across the image to occupy all possible positions. At each position the weight values of the filter are multiplied with the corresponding pixel values, ultimately producing a transformed image representation in which the pixel values that match the pattern of weights in the filter are amplified. The transformed images are normalized and passed to the

pooling layer, which repeatedly steps a small mask (not necessarily the same size as the filters) across the image and for each position extracts a single value from the area of the mask, typically the maximum or average value of the contained pixels. This action creates an array called a feature or activation map, smaller than the original image, in which the presence and relative locations of the amplified values (“hot spots”) produced by the filters are preserved, and noncontributing details are eliminated. The feature map is passed through several additional convolution and pooling layer pairs, which repeat the filtering and pooling process to produce progressively smaller and more abstract feature maps. Although the first feature map highlights basic image details, such as edges, corners, colors, or color gradients, subsequent feature maps represent groupings of details into important higher-level motifs, and then groupings of these into collections that represent characteristic elements defining the image class. The final feature maps are compact, abstract representations of the original image, and they are used as the input features into the feed-forward portion of the network. This section of the network uses the input features to determine the probability that an image belongs to a particular class (Figure 4, D). The number and size of the filters, the pattern of filter movement across the image, and the size of the pooling masks are set manually prior to training the network. Training a CNN may require tens to hundreds of thousands of images.¹⁷ As the network is trained across the many images, the patterns of weights in

the filters are progressively modified by backpropagation such that filters become able to highlight groupings of pixels and motifs important for correct image classification, and the feed-forward network parameters are modified to correctly classify the images based on those features.

Because CNNs produce feature maps that can be traced back to locations in an image, they have the potential to be less opaque than some machine learning techniques. *Saliency maps*⁸⁵ are representations of the feature maps that are overlaid on the original image and indicate which portions of an image a CNN has identified as containing important motifs. This display allows the feature identification portion of the CNN to be checked for face validity, for example, whether the CNN is using portions of the image that are believed to be important for classification. However, saliency maps are limited in that they do not identify the specific image details within the identified area of interest that are used, and they do not express how various motifs are weighted in the feed-forward part of the network that makes the final classification.

Recurrent neural networks^{83,86,87} are designed to work with sequence data in cases where there is important information in the sequential values of data elements. Recurrent neural networks have been used for interpretation, translation, and creation of speech and text, including automated report generation in radiology.⁸⁸ They are also capable of time series analysis and are useful for anomaly detection in quality control data.⁸⁹ There are a number of variations of recurrent neural network architecture, but the essential feature is storage of data from prior instances to use as input for subsequent instances (Figure 4, E), providing contextual information from past input in addition to the current input. Network nodes may accumulate historical data by adding their output to prior output values, storing the result, and using it as input for the next instance. The stored value, or context node, has its own weighting that is adjusted by backpropagation during training. A variation of a recurrent neural network called *long short-term memory* (LSTM) has proven to be particularly capable. Its data storage units are called gated nodes, and they can control their contents and activity based on incoming data and weights that are defined during backpropagation. Gated nodes have the ability to capture, preserve, and weight data from arbitrary instance sequences that may be remote or discontinuous in time, and thereby flexibly represent short- or long-term dependencies in sequences. An LSTM network learns during training what and how much past and current data to store for optimal performance.

Generative Adversarial Networks (GANs; not shown) are pairs of neural networks that are trained in tandem to create data with particular characteristics.^{90,91} One member of the pair is a discriminator network that is trained as a classifier, and the second is a generator network that is trained to produce synthetic instances that will fool the classifier. In an imaging context, the discriminator is a convolutional network that determines whether an image is synthetic, and the generator is a deconvolutional network that works like a convolutional network in reverse to produce synthetic images that have features similar to those of the discriminator training set. During GAN training the output of each network is used to further train the other network, as the generator network progressively becomes better at producing images that have characteristics of the original discriminator training set, and the discriminator progressively becomes better at distinguishing the synthetic images

from the real ones. Generative Adversarial Networks have been used to create synthetic histologic images,⁹² propose useful new chemical structures in drug development,⁹³ and define protein structure from sequence.⁹⁴ However, they are best known to the public for producing photorealistic images and videos of nonexistent humans or events (DeepFakes⁹⁵). Generative Adversarial Networks are also one method for creating adversarial images, which are discussed later.

This discussion focuses on supervised neural networks because they are likely to be important for pathology applications during the next several years. Unsupervised applications of neural networks, such as self-organizing maps, are beyond this scope and have been reviewed elsewhere.⁹⁶ There are many other machine learning algorithms that are very useful in settings for which they are suited, including decision-making with laboratory and electronic health record (EHR) data. Although these algorithms may not perform as well with high-dimensional data as neural networks trained optimally with large data sets, they may outperform neural nets with certain types of data or when the amount of training data is constrained (Figure 4, C). A selection of traditional machine learning algorithms grouped by type are listed in Figure 2 and described below, and displayed in Figure 5.

OTHER REGRESSION ALGORITHMS

The simplest regression algorithm is *linear regression*, which uses a linear equation as a model and finds the line for which the deviations of a set of known data points are minimized. The fitted equation can then be used to predict the value of an outcome variable based on an input variable (Figure 5, A). *Multiple linear regression* expresses the relationship between a single outcome variable and 2 or more input variables simultaneously. For example, new versions of the Magee Equations⁹⁷ use multiple linear regression based on relatively low cost pathology data (Nottingham score, Ki-67 index, tumor size, H-scores for estrogen and progesterone receptors, and human epidermal growth factor receptor) to predict the Oncotype DX score, a more costly genomic analysis for determining the likely benefit of chemotherapy and risk of recurrence in breast cancer. *Polynomial regression* models nonlinear data using exponentials of the input variables (Figure 5, B). Higher-order polynomial regression tends to be sensitive to noise in the training data and may produce complex models yielding results that do not generalize to nontraining data (ie, “overfitting” the training data; see below). *Regularization methods* (not shown) mitigate this problem by placing constraints on the model’s variability in the setting of noisy data.⁶⁷ For example, *ridge regression* modifies the loss function to avoid changes in weights during training that do not yield a required minimum performance improvement. *Lasso regression* and the *Elastic Net* method modify the loss function to limit weighting changes and also force the weighting of less important features to near zero. Elastic Net is particularly useful in cases where there is a large number of features per instance compared with the number of instances in the training set, or when some features are correlated.

OTHER CLASSIFICATION ALGORITHMS

The simplest classification method is *logistic regression*, which yields a binary classification.⁶⁷ Like linear regres-

Other regression and classification algorithms

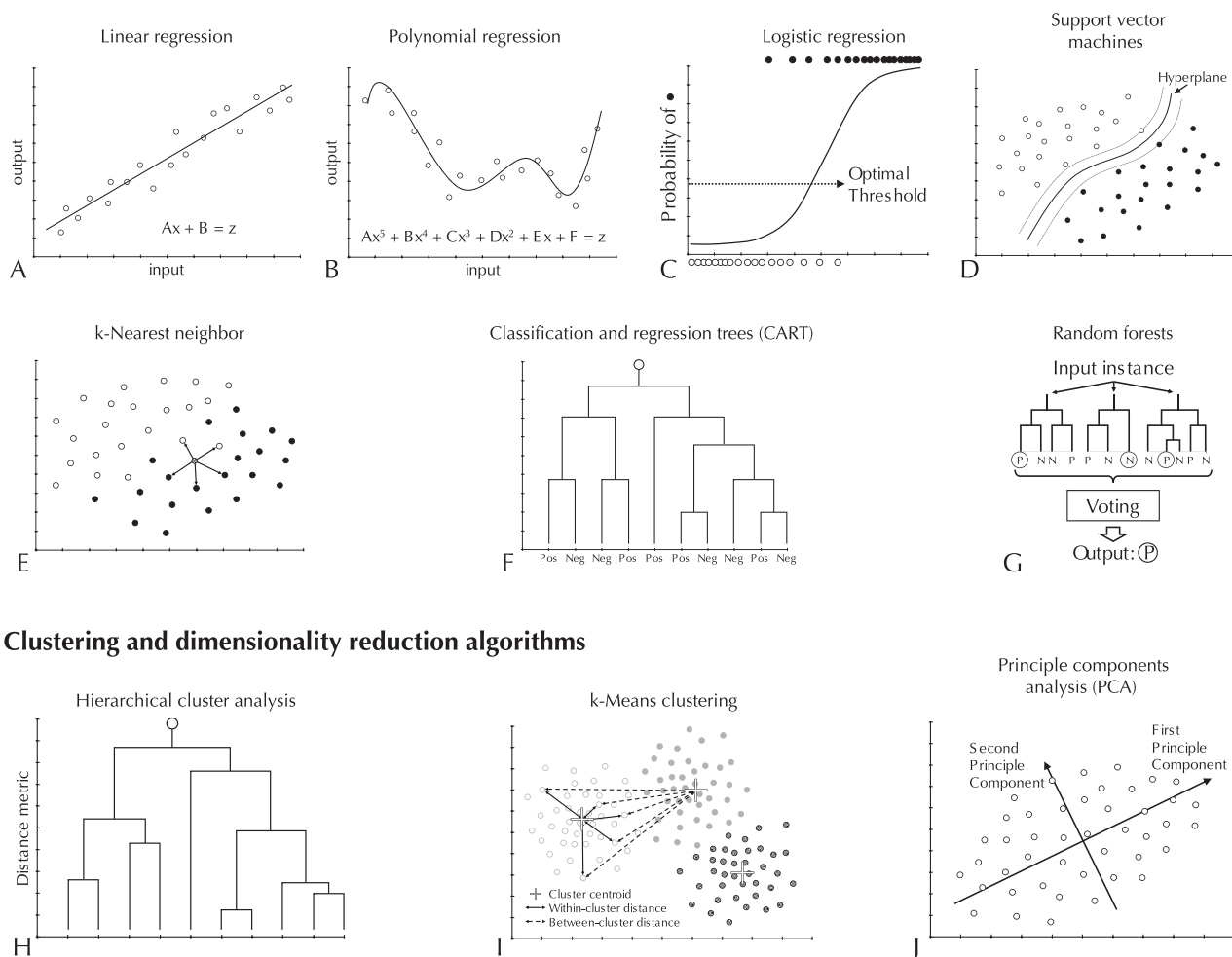


Figure 5. Machine learning algorithms other than neural networks. Linear regression (A), which fits a straight line to input data to predict an output value, is the simplest supervised regression algorithm. Polynomial regression (B) fits an exponential polynomial to input data to predict an output value and accommodates nonlinear relationships. Logistic regression (C) is a simple classifier based on a sigmoid relationship that computes a probability of class membership with a defined threshold for class assignment. The distributions of classes (closed and open circles) associated with the input data are shown above and below the graph. A Support Vector Machine (D) computes a boundary (the hyperplane) between the closest members of different classes, such that the boundary margins are maximized. k-Nearest Neighbors (E) plots all training instances in multidimensional space and classifies new instances based on the identities of the closest training instances. Decision trees, such as CART (F), classify instances based on optimized sequences of decisions with threshold values. Random forests (G) creates many different short decision trees and classifies instances based on the aggregate output of the trees. Hierarchical Cluster Analysis (H) is an unsupervised algorithm that progressively joins instances into larger and larger clusters based on a chosen distance metric, growing dendrograms such as the one shown here from the bottom up. k-Means clustering (I) is an unsupervised method for assigning instances to a defined number of clusters by iteratively calculating the cluster centroid locations and switching instance memberships to the closest centroid. Principal Components Analysis (J) defines a new coordinate system with orthogonal axes that are aligned to the directions of greatest variability in the data.

sion, logistic regression optimizes the parameters of a simple function to fit the data, but in this case the model is the logistic function, which yields a sigmoid output and calculates the probability of membership in the default class. A threshold applied to the probability yields the binary class assignment (Figure 5, C). This strategy of calculating a probability or other class membership score with a threshold for class assignment is also used in more complex classifiers. Logistic regression has been used extensively to develop clinical prediction models,⁹⁸

including prediction of sepsis from EHR data,⁹⁹ and is often used as a baseline model to which the performance of other classifiers is compared. It has limitations in handling large numbers of input variables, correlated input variables, and situations where the mathematical relationship between the input variables and output is not constant.¹⁰⁰

The *Support Vector Machine* algorithm is mostly commonly used for supervised binary classification.¹⁰¹ It determines the optimal boundary between the 2 classes in multi-dimen-

sional space in which each feature is a dimension. This boundary is called a “hyperplane” and its position is defined during training by optimizing its margins as it passes between instances of different classes. In the case of instances with 2 features, the hyperplane is a line and can be visualized (Figure 5, D). When instances have more than three features, the hyperplane cannot be easily visualized, but the mathematics is analogous. After training, new instances are classified based on their location with respect to the hyperplane. Support Vector Machines are relatively efficient in terms of computer memory and tend to work well with high dimensional data (ie, many features) as long as the number of features does not exceed the number of instances. Support Vector Machines have been used in fetal aneuploidy screening,¹⁰² prediction of metastasis from gene expression profiles,¹⁰³ and autoverification of gas chromatography mass spectrometry assays in the clinical laboratory.⁵⁸

The *k*-Nearest Neighbors (k-NN) algorithm also plots a population of instances in a multidimensional space in which each feature is a dimension, but it is an instance-based method and thus does not train a model to calculate an outcome for new instances.¹⁰⁴ Instead, it defines a distance metric that determines the nearest neighbors to a new instance when that instance is plotted into the population (Figure 5, E). In classification problems the new instance is assigned the most common class among the neighbors. A k-NN can also be used for regression problems, in which the new instance may be assigned the mean of the dependent variable among the neighbors. In k-NN the “knowledge” of the system is stored in the structure of the population data rather than in a trained model, and the system can be updated incrementally by adding vetted new instances to the population as they become available. The form of the distance metric and the number of neighbors considered in evaluating new instances (“k”) are manually defined. Variations of k-NN allow neighbors to be weighted by distance or other characteristics. A k-NN approach has been used to derive tumor infiltrating lymphocyte density in breast cancer¹⁰⁵ and to determine whether tumor infiltrating lymphocyte density could independently predict pathologic complete response.¹⁰⁶

*Decision Trees*⁶⁷ are a flexible method commonly used for supervised classification but also broadly useful in other settings (Figure 5, F). A common form is called Classification and Regression Trees, or CART. CART defines if-then decisions and cutoff values that are applied in an optimal sequence to classify a set of training instances as correctly as possible. Decision points are referred to as “nodes,” and trees extend from a root node through internal nodes creating branches to a leaf node at the end of each branch. During training (or “growing” decision trees) using a population of known instances, at each node the algorithm chooses the feature and cutoff value that splits the population into the purest possible subsets of classes, and then recursively does the same for subsequent nodes until class purity cannot be increased. Class purity is measured by *Gini impurity* or *entropy*, which are zero with single-class populations and high when there is a large number of evenly mixed classes. These metrics are minimized during training in a manner analogous to a loss function in other algorithms. The root and internal nodes represent the independent variables (the features), and the leaf nodes represent the dependent variable or outcome (the class). Once the tree is grown, new instances can be classified by

putting them through the tree’s sequence of decisions, which yields a probability of class membership based on the composition of the training subset for the leaf node at which the instance arrives. Decision trees are susceptible to creating branches that are not useful in response to small variations in data (ie, overfitting, see below). This problem may be addressed by limiting the number of nodes allowed in a branch or by adding only nodes that produce a statistically significant increase in purity in their branches (forms of “pruning” that simplify a tree). Variations in the algorithm have been developed that allow for more than 2 outcomes per node, support regression using the average value of instances in the training population assigned to a leaf node, and assign different weights for false-negative or false-positive results during training so that a tree can be optimized for use in either a screening or confirmatory context. The nodes of a decision tree are easy to express as if-then statements with cutoff values, and therefore decision trees are transparent and relatively explainable in operation, in contrast to a number of other machine learning algorithms. Decision trees have been used for molecular subtyping of breast cancer¹⁰⁷ and expression of clinical guidelines for breast cancer management in executable form.¹⁰⁸

CLUSTERING AND DIMENSIONALITY REDUCTION ALGORITHMS

Clustering algorithms are typically unsupervised and yield information about the relationships among instances and their features (typically called dimensions in this context) in a population. Rather than a predefined “correct” target for assignment or prediction, such as a dependent variable or class label as in supervised learning, the goal of unsupervised learning is to reveal patterns in the data, such as groupings of instances that have maximum commonality among their dimensions. This knowledge can be useful in creating hypotheses about structure and associations in data sets, or in simplifying a data set while preserving its most important features.

*Hierarchical Cluster Analysis*¹⁰⁹ is widely used in bioinformatics, genetics, and other fields to reveal relative similarities between members of a population of instances. The most common approach is agglomerative clustering, in which each instance of a population starts as its own individual cluster, and then clusters are progressively joined based on a distance metric and linkage criteria until all instances are members of a single cluster. Instances or clusters that are more similar are joined sooner than less similar ones, and the sequence of cluster aggregation, typically expressed by a dendrogram (Figure 5, H), reveals the similarity relationships. Clustering is carried out in a multidimensional space in which each feature of the instance population is a dimension. At each step, the algorithm identifies and joins the 2 clusters that are closest. In one common variation (Ward method), the closest clusters are identified as the 2 for which merging produces the smallest increase in within-cluster variability. Variability increase is measured by a loss function that subtracts the combined squared distances between cluster instances and the cluster center (the sum of squared errors [SSE]) for the initial 2 clusters from the SSE for the merged cluster. There are a number of other variations with different distance metrics and linkage criteria, and the choice of the clustering

Table 1. Selected Metrics Useful in Evaluating Machine Learning Performance^a

Metric	Application	Definition
Root mean squared error (RMSE)	Regression	Square root of the average squared error from predicted, with greater weight for larger errors
R squared (R^2)	Regression	The proportion of the dependent variable's variation that is produced by the independent variable
Accuracy	Classification	Proportion of all predictions that are correct, may not perform well with unbalanced classes
Sensitivity (Recall ^b)	Classification	Proportion of class members correctly identified (true positive rate for the class)
Positive predictive value (Precision ^b)	Classification	Proportion of positive results that are correct
Specificity (Selectivity ^b)	Classification	Proportion of non-class members that are correctly identified (true negative rate for the class)
F1 Score	Classification	Harmonic mean of recall and precision, with equal weighting of each
Matthews Correlation Coefficient (MCC)	Classification	Overall correlation of observed and predicted binary classification, incorporates true and false positives and negatives; useful in class imbalance settings
C-statistic	Classification	Concordance statistic; the area under the ROC curve; independent of any particular discrimination threshold
FROC true-positive fraction (TPF)	Classification	True-positive fraction at a specified false-positive rate, used with FROC curves
Hosmer-Lemeshow test	Calibration	$P < .05$ indicates a difference in observed versus predicted class probabilities and poor classifier calibration; frequently used but not recommended because of low power and difficulty in interpretation ¹³¹
Sum of squared error (SSE)	Unsupervised clustering	Sum of squared distance between each member of a cluster and the cluster's center; a measure of cluster compactness
Silhouette coefficient	Unsupervised clustering	Similarity of members of a cluster to that cluster versus the next nearest cluster; a measure of cluster definition
Calinski-Harabasz index	Unsupervised clustering	Ratio of dispersion between clusters to dispersion within clusters; a measure of overall clustering consistency
Cross-validation error	Validation	Variation in performance between cross-validation folds; a measure of overfitting

Abbreviations: FROC, free-response receiver-operating characteristic; ROC, receiver-operating characteristic.

^a Data derived from primary sources.^{123,125,140,174–176}

^b In the information retrieval domain, the terms sensitivity, specificity, and positive predictive value are replaced by recall, selectivity, and precision, respectively. This review follows clinical laboratory terminology, but machine learning literature often uses the information retrieval terms.

approach should be based on the form of the data and the goals of the analysis.¹⁰⁹

K-Means clustering (Figure 5, I) is an unsupervised algorithm that assigns instances from a population to a manually specified number (“K”) of clusters based on their similarity.¹¹⁰ The algorithm initially assigns all instances randomly to one of the clusters in multidimensional space, in which each feature of the instances is a dimension. It then iteratively calculates the position of the center (centroid or average) of each cluster, compares the positions of all the instances to the centroids using a defined distance metric, switches instances to the cluster with the closest centroid, and recalculates the centroid positions. Processing stops when the centroids stop moving and there are no more cluster switches among the instances. Ideal grouping will not occur if the specified number of groups (K) does not fit the characteristics of the population. Optimal numbers for K can be estimated by comparing the compactness of groups for a range of K numbers. The lowest K consistent with compact groups as measured by cluster sum of SSE (Table 1) or a ratio between cluster SSE and cluster separation (ie, the silhouette coefficient; Table 1) is optimal.¹¹⁰ K-means clustering has been used to measure the relative increase in number of nuclei in digital histology images of normal cervical squamous epithelium versus different grades of cervical intraepithelial neoplasia as part of an algorithm for cervical intraepithelial neoplasia classification.¹¹¹

Cluster analyses may involve instances that have many dimensions, and the values of some dimensions may be correlated. Correlated or redundant dimensions do not individually provide unique information useful in clustering, but they do increase the complexity of the analysis and may introduce noise that makes cluster separation more difficult. Hence, their identification and removal often improve cluster and other types of analysis by simplifying a data set while preserving its essential characteristics. Unsupervised machine learning algorithms that rank the importance of dimensions are collectively called *dimensionality reduction methods*. *Principal Components Analysis* is a commonly used method that aggregates dimensions and ranks them based on their contribution to the variability in a population.⁶⁷ The approach can be understood as plotting the entire population in a multidimensional space (as in the clustering methods above), and then defining a new coordinate system in which the axes align with the maximum variance in the cluster and are orthogonal with each other (Figure 5, J). These axes are referred to as principal components and together account for all the variability in the data set, in decreasing order from the first to the last. Because they are orthogonal, the variability accounted for by each component is uncorrelated with the other components. Although dimensions beyond 3 cannot be easily visualized, their mathematics and representation of variation is analogous to the 3-dimensional case. In multidimensional contexts there

are as many principal components as dimensions, with the last several components usually contributing very little to overall variability. In its basic form principal components analysis requires continuous numerical dimensions, but variations have been developed to incorporate categorical dimensions. Principal components analysis can be used before hierarchical or k-means clustering to identify unimportant dimensions for exclusion, or the first several principal components themselves can be used as the dimensions for clustering (PCA Clustering), which may yield more compact and better-separated clusters than the native dimensions.¹¹²

ENSEMBLE METHODS

In some cases, the aggregated output of a group of relatively simple models with weak individual performance but diverse perspectives on the data can outperform a single more complex model. Machine learning strategies that rely on groups of models are called ensemble methods.¹¹³ The models may be trained in parallel (independently), or they may be trained in sequence, with the output of one model affecting the training data of the next model.

The models in *parallel ensembles* may be of similar or different types. If the models are similar, it is important to create diversity among the models or the ensemble will have no advantage compared with a single model. One way to create diversity is to train the individual models against random subsets of training instances, a process known as *bagging* (for bootstrap aggregating).¹¹³ A second approach called the *random subspaces* method is to train each model against all training instances but use random subsets of instance features per model, which is particularly useful when the feature set is large, as with images. Random Forests (below) uses both of these methods. When models are of different types, their different training algorithms naturally introduce diversity, but these methods can still be used if additional diversity is needed. The output of parallel ensembles for classification problems is usually aggregated by voting. Setting the output to the most frequent class output is referred to as “hard voting.” If all the models can output class membership probabilities, best performance is often achieved by averaging the probabilities and setting the output to the class with the highest result, so-called soft voting. In regression problems the output of the ensemble is set to the average output of the individual models.

Sequential ensembles are typically constructed using *boosting* methods,^{113,114} which sequence the training of a set of models such that incorrect output produced by models trained earlier is mitigated by models trained later, leading to strong overall performance. The most popular strategies are Adaboost (short for Adaptive Boosting) and Gradient Boosting. *Adaboost* is designed for ensemble classifiers and typically uses multiple short decision trees as weak models. As training by supervised learning is completed for each model in the sequence, misclassified instances are weighted (boosted) for the next model, increasing the likelihood that those instances will be classified correctly. The sequence of models grows until performance of the aggregate does not improve or until a defined number of models is reached. When new instances are evaluated, the vote of each model is weighted based on its overall accuracy on the training data, in contrast with the parallel methods discussed above, where the votes are weighted equally. Adaboost was originally developed for binary classification but has been

extended for use with multiple classes and class probabilities analogous to “soft voting” in parallel methods. *Gradient Boosting* uses a loss function to calculate the errors of the previous model in the training sequence, and then trains the next model on this data so that its output reduces the error. The aggregate output across the set of models progressively approaches optimal as each new model is added, analogous to gradient descent optimization of individual models discussed previously. Classification or regression with new instances is accomplished by weighted voting similar to Adaboost or averaging, respectively.

The *Random Forest* method is a highly effective and widely used classification algorithm that combines the output of hundreds to thousands of simple decision trees (Figure 5, G).⁶⁷ The method uses several strategies during training to create randomly varied trees that each have slightly different perspectives on a data set, and for new instances the model returns the class probability determined by a vote of the population of trees. Variation is produced by growing trees using random subsets of a training population of instances and by limiting searches for the best splitting strategy at tree branches to random subsets of instance features. The trees are kept simple by limiting the number of nodes in branches (ie, keeping the trees short). The variability and simplicity of the trees mean that individually they do not perform as well as a fully optimized single decision tree, but in aggregate they are able to respond more flexibly and accurately in classifying new instances, and there is less of a tendency to overfit. An additional useful aspect of the method is that it is possible to determine the relative importance of individual features for a classification task by calculating a weighted average of the reduction in class impurity produced by all nodes in the forest that use each feature. Random forests have been used in predicting 30-day readmissions based on EHR¹¹⁵ or insurance claim¹¹⁶ data, and for distinguishing non-small-cell lung cancer from other forms of lung cancer by radiologic findings.¹¹⁷

MODEL DEVELOPMENT AND TRAINING

Machine learning software may be written in any complete programming language, although the core routines that implement demanding tasks such as model training are usually written in a compiled language for speed and efficiency, and made available as open source¹¹⁸ or commercial software libraries that are accessible from a variety of software development environments. These libraries implement many standard machine learning algorithms and provide prebuilt workflows and software wrappers for configuring and training models. The most common languages for experimentation are *Python* and *R*, which are open source, and both languages have a large supporting infrastructure of open source machine learning libraries. Large technology companies with strong interests in AI contribute comprehensive code libraries, such as TensorFlow (Google, <https://www.tensorflow.org>; accessed October 7, 2020) and PyTorch (Facebook, <https://pytorch.org>; accessed October 7, 2020), to the open source community. Although much of the current academic work with machine learning is based on open source software, commercial analytics companies such as SAS (<https://www.sas.com/>; accessed October 7, 2020) and MathWorks (<https://www.mathworks.com>; accessed October 7, 2020) also produce high-quality machine learning development tools.

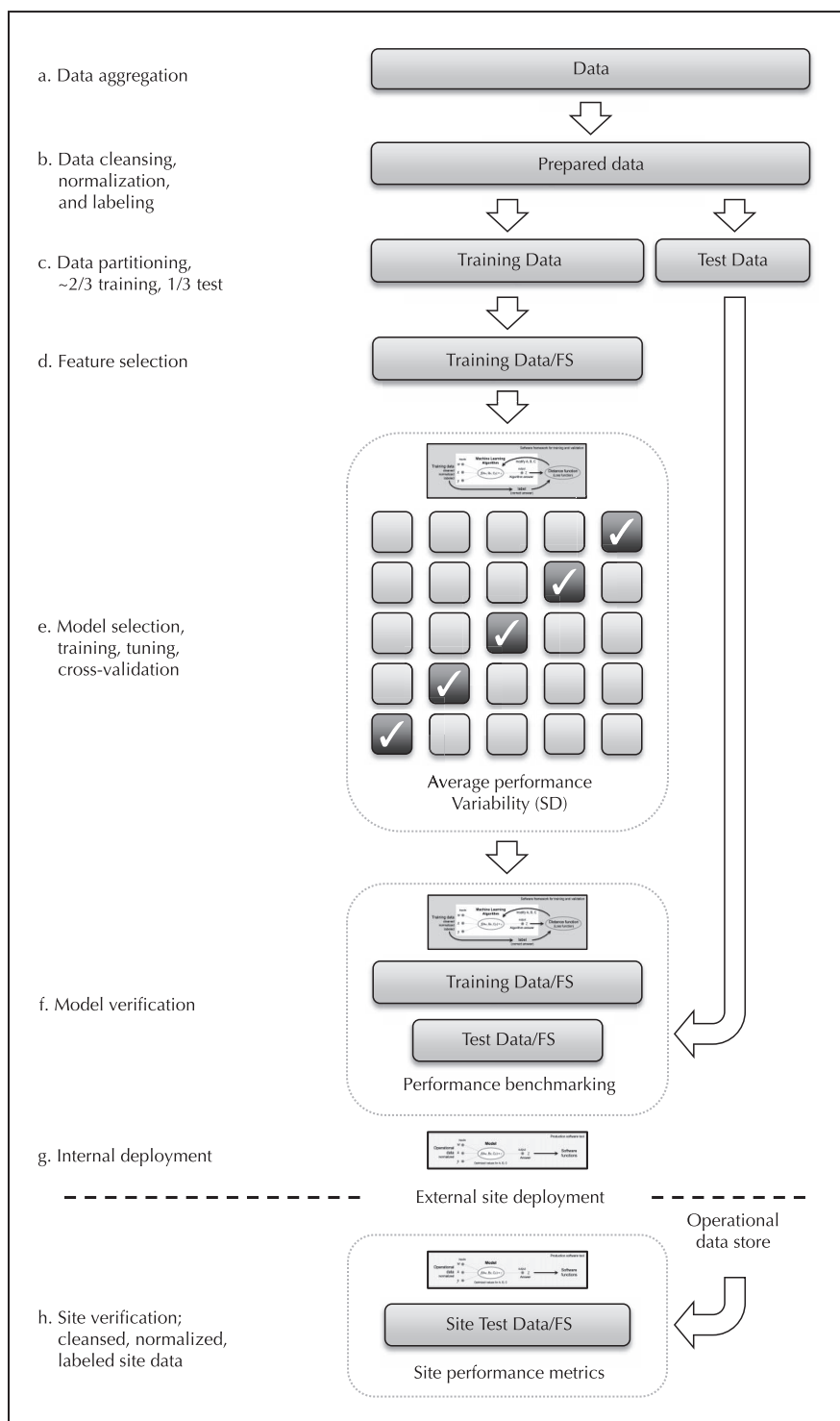


Figure 6. Model development and deployment pipeline. Prior to model development, data are acquired (a), cleaned and normalized (b), and partitioned into training data and held-out test data (c). Development data must be representative of the operational data that the model will eventually use. Feature selection (FS) determines which instance data will be used for machine learning (d). The model is trained and optimized (see Figure 3, Training) on the selected features of the training data using cross-validation (e). The figure depicts 5-fold cross-validation in which the training data are split into 5 segments and the model is trained and tested 5 times, with each data segment serving as the test set (check mark) once. Finally, the model is trained on all the training data using the optimized settings and verified against the held-out test data using the selected features (f), and its performance is compared with the previous validation performance as a benchmark. After verification, the model can be incorporated into software (g) and deployed (see Figure 3, Deployment) at the development site. If the model is to be deployed at other sites that did not provide training/test data, the model must be reverified against operational data at each deployment site using the same features as the training data to confirm generalizability with adequate performance (h).

The most useful machine learning software in pathology in the near term will likely be based on supervised learning models. An overview of the training and deployment process for these types of models is shown in Figure 6. The general elements required for machine learning model development^{66,119} are similar to those for statistical predictive models.^{98,120} With appropriate expertise and access to adequate amounts of data, it is possible for laboratories to develop models for local use, similar to in-house developed laboratory tests. In this case, the laboratory would carry out

the processes shown in Figure 6, a through g, leading to internal deployment. However, most machine learning software for medical use in the future will probably be based on models developed commercially using large aggregates of data from multiple sites, and then deployed at external client sites with local verification. In that scenario, the developer would carry out the processes in Figure 6, a through f, and the local site would be responsible for external deployment and the final local verification step, with developer support (Figure 6, h). The specific form and

regulatory requirements for this local verification have not been established and are currently under active discussion (see below).

The first step in developing a supervised model is defining a task appropriate for a supervised learning application. This might be a classification or prediction task that was previously executed using expert judgment, and that involves multiple data elements without well-defined relationships. Example data relevant to the task are assembled (Figure 6, a), ideally from multiple sources if the model is to be generalizable. The required volume of data for primary development could range from hundreds to millions of instances depending on the nature of the task and the type of model. For supervised learning, the training instances must have a defined class label (such as a diagnosis) or dependent variable value that is accurate. Label assignment may need to be carried out manually for each training instance, which can be laborious. Several methods for reducing the requirement for manual labeling in supervised learning were discussed above (see Types of Machine Learning). Each instance should have an equivalent set of features, and all features should have similarly scaled values. Feature data are typically normalized to standard deviation units (if normally distributed) or a percentile span of the data range (if not). Normalization and scaling of features to prevent overemphasis of higher magnitude features is particularly important for methods that use distance or similarity metrics. If there are missing or erroneous data they should be corrected or values should be imputed that are statistically appropriate for the learning task. Instances with unreasonable data (outliers) may need to be removed from the population if that can be done without introducing bias. The model will perform best if value distributions of the features in the final training data set reflect the real-world data distributions that the model will see in routine use. The process of examining, correcting, and completing the data can be time-consuming and is often referred to as “data cleansing” (Figure 6, b). The feature data should not be pristine because they should resemble real-world data in terms of noise, distribution, and overall quality, but the labels or dependent variables must be accurate. After cleaning and labeling, the data are partitioned randomly into a training set that is used to build the model and a test set that is held back and used to measure the performance of the model after training is complete (Figure 6, c). Generally, about two-thirds of the original data is used for training, and one-third is held back for testing, and both data sets should have similar proportions of classes and distributions of feature values.

Features may differ in their importance for a machine learning task, and identification of the important features for inclusion in training is called *feature selection* (Figure 6, d).¹²¹ Using only important features speeds up model training by reducing the amount of data that must be processed, and it improves the performance of many models by reducing the likelihood of overfitting on irrelevant data⁶⁶ (see below). Feature selection is sometimes based on choice by human experts, although this approach may not be optimal when many features may be relevant or there may be unknown relationships. Features may be evaluated independently using univariate methods to identify those that have statistically significant relationships with the class label or dependent variable, although this approach may miss useful contributions from less impactful features or patterns of multiple features. At the cost of additional processing, a

simple machine learning model may be trained with varying combinations of features, ranking features by their effect on model performance across the combinations. Random forests, as noted above, can return the relative contribution of features to its voting patterns and thus reveal the relative importance of features. This type of trial with a simple model can help to identify features that are likely to be useful with any machine learning algorithm and may yield feature groups with better performance than univariate analysis.⁶⁶ Finally, feature selection is built into some machine learning algorithms and operates as part of the primary model training, automatically driving the weights of unimportant features to near zero. Examples of the latter include ridge regression, lasso regression, and Elastic Net, discussed above, and deep neural networks perform similarly if given large amounts of training data.

Machine learning models are trained by optimizing defined performance metrics. A variety of general and specialized metrics are available,^{122–125} and the choice of metrics for optimization depends on both the type of model and the type of task the model will carry out. Commonly used metrics for regression models measure how closely predicted values match actual values, for example, *root mean square error* and *mean absolute error*. Metrics for classification models measure discrimination between classes based on a score or probability for each class compared with a threshold. Threshold-dependent metrics include *accuracy*, *sensitivity* (recall), *specificity* (selectivity), *positive predictive value* (precision), and *F1 Score* (Table 1). The *Matthews Correlation Coefficient*, which takes into account true and false positives and negatives, can be used when classes are unbalanced and is an example of a more specialized metric that has been employed in pathology image classification.¹²⁶ Additional useful performance metrics are listed in Table 1.

Once the data are prepared, features are selected, and metrics are chosen, the model is trained using the training data set (Figure 6, e). At the start of training, weights and other learnable model parameters are initialized, typically to random numbers. As the model makes errors on the training data, the parameters are adjusted by the training algorithm to improve model performance as assessed by the defined metrics. Early in training, the model performs poorly on the training data, and if it were to be tested against the held-out set of test data it would also perform poorly (Figure 7, A and B). At this stage the model is “underfit” with respect to the training data because it does not fully reflect the underlying relationships in the data. With successful training the performance of the model improves until it reaches an optimal point where it performs well on both the training and test data (Figure 7, A and C). At this point the model reproduces underlying relationships in the data as closely as possible given its inherent limitations. If training is continued, model performance can continue to improve on the training data but decline on the test data (Figure 7, A and D). This is because the model is beginning to incorporate minor patterns in the noise of the training data that do not reflect the true underlying relationships in the data. When this occurs, the model is “overfit” with respect to the training data. Overfitting is a risk in many machine learning projects and can be mitigated by comparing model performance on the training and test sets as training progresses (as in Figure 7, A), assessing performance variance using bootstrapping or k-fold cross-validation (see below), and maximizing the volume of training data.

The effectiveness of training is influenced by the way the training session is carried out, and these details are specified by parameters that are not learned but are manually set before training. These parameters, commonly called *hyperparameters*, define, for example, the amount of change in input weights that the loss function can make per training step, limits on the input weights, the total number of “epochs” (passes through the training data) to be carried out, and allowable model complexity within the constraints of a particular algorithm, for example, the number of decision nodes and leaves allowed for decision trees, the number of layers and nodes per layer in a deep neural network, and the number of clusters in k-means clustering. Although hyperparameter settings substantially affect the performance of a model there is in most cases no theoretical basis for choosing particular values. Hence, model performance for alternative hyperparameter values is often tested in multiple training runs by an exhaustive combination of defined sets of values, random selection of values from a defined statistical distribution, or probabilistic search methods starting from a random sampling and focusing testing around parameter combinations that perform well.¹²⁷

Preliminary performance evaluation during selection of models and model optimization is usually carried out using *bootstrapping* or *cross-validation* techniques to assess overfitting. In bootstrapping, multiple preliminary training sets are created by random selection with replacement from the training data, the model is trained on each set separately and tested against the instances that were not chosen for that set, and the mean and standard deviation of the performance metric are calculated across the sets. These values are predictive of real-world performance, and significant performance variability across the bootstrap sets suggests that a model may be overfit and may produce poor and inconsistent results if deployed. *K-fold cross-validation* is a related technique that divides the training data into k equal subsets (folds) with typical k values between 5 and 10. In cases where there is significant class imbalance, class proportions should be held constant across the folds (*stratified k-fold cross-validation*). Figure 6, e illustrates 5-fold cross validation. Training is carried out on 4 of the folds combined, and the fifth is used to test the trained model. That process is repeated 4 more times until each fold has served as the test fold. K-fold cross-validation has the advantage of using all the instances in the training data for training and preliminary testing. As in bootstrapping, the mean and standard deviation of performance across the folds is predictive of deployed performance, and high variability suggests overfitting and a likelihood of poor and inconsistent performance. If performance is poor, model hyperparameters may be adjusted, data quality and processing may be reviewed, or a different type of model may be considered. If performance is acceptable, the metrics calculated during bootstrapping or cross-validation establish a benchmark for performance at verification after final training.

In the final step of development, the model with its optimized settings is trained using the entire training data set, and its performance is verified against the held-out test data set (Figure 6, f). A well-trained model should perform against the test data similarly to its average performance against the training data in the bootstrapping or cross-validation steps above; poorer performance suggests overfitting during training. Performance of regression models may be expressed as *root mean square error*, *correlation* (R^2 or

adjusted R^2), and *calibration* (Table 1). Root mean square error and correlation measure the overall agreement of predicted values with true values, whereas calibration is analogous to linearity in traditional laboratory tests and represents the agreement between predicted and true across the operating range. Classification performance is expressed as discrimination and calibration.¹²⁸ *Discrimination* assesses the accurate separation of classes. Basic threshold-dependent classification metrics were discussed previously (Table 1). Overall discrimination capability independent of any particular threshold is commonly evaluated using area under the receiver-operating characteristic (ROC) curve (Figure 7, E), also called the concordance statistic or *C-statistic* (Table 1). The ROC curves for machine learning classifiers are created by varying the threshold for class assignment across the operating range of the classifier and plotting the resulting values of sensitivity versus false-positive rate. Area under the curve (AUC) values approaching 1 denote high discrimination performance. Discrimination in image classification is sometimes evaluated using free-response ROC (FROC) curves, which are roughly analogous to ROC curves but accommodate multiple false positives or negatives per image or slide (Figure 7, F, see additional discussion below). *Calibration* for classifiers refers to the accuracy of class membership probability prediction across the range of reported probabilities. Good calibration is crucial for prognostic models where medical treatment decisions rely on the probability that an individual patient is a member of a particular class or has a particular outcome risk.^{119,129,130} Models with good overall discrimination (well-placed class assignment thresholds) may nevertheless show poor calibration (inaccurate actual probability predictions).¹³¹ Classifier calibration may be assessed using reliability diagrams, which plot the observed proportions of a class versus its predicted probability across the operating probability range (Figure 7, G).^{128,131,132} Several metrics have also been used to express the adequacy of calibration, including the slope and intercept of the reliability diagram (Figure 7, G), and the overall observed versus predicted (observed/expected) ratio, which expresses average calibration across the operating range, albeit with less useful detail than the diagram.

The model development process described above can be considered a processing pipeline beginning with data assembly and continuing through feature selection, model selection, model optimization, and verification. This pipeline can be standardized and automated in high-performance computing environments. Prototype software frameworks have been developed that automate data processing, feature selection, and model training using methods chosen by the software based on data characteristics, and automatically identify the best-performing models for optimization, verification, and reporting. Although most models are currently developed interactively by human experts who manage and review each stage of the pipeline, automation of machine learning is, perhaps ironically, an active focus of machine learning research.¹³³

ADDITIONAL CONSIDERATIONS IN PERFORMANCE ASSESSMENT

Machine learning models are susceptible to 2 types of error, bias and variance, which are analogous to accuracy and precision in the clinical laboratory. *Bias* refers to the fidelity of representation of the true relationships in the

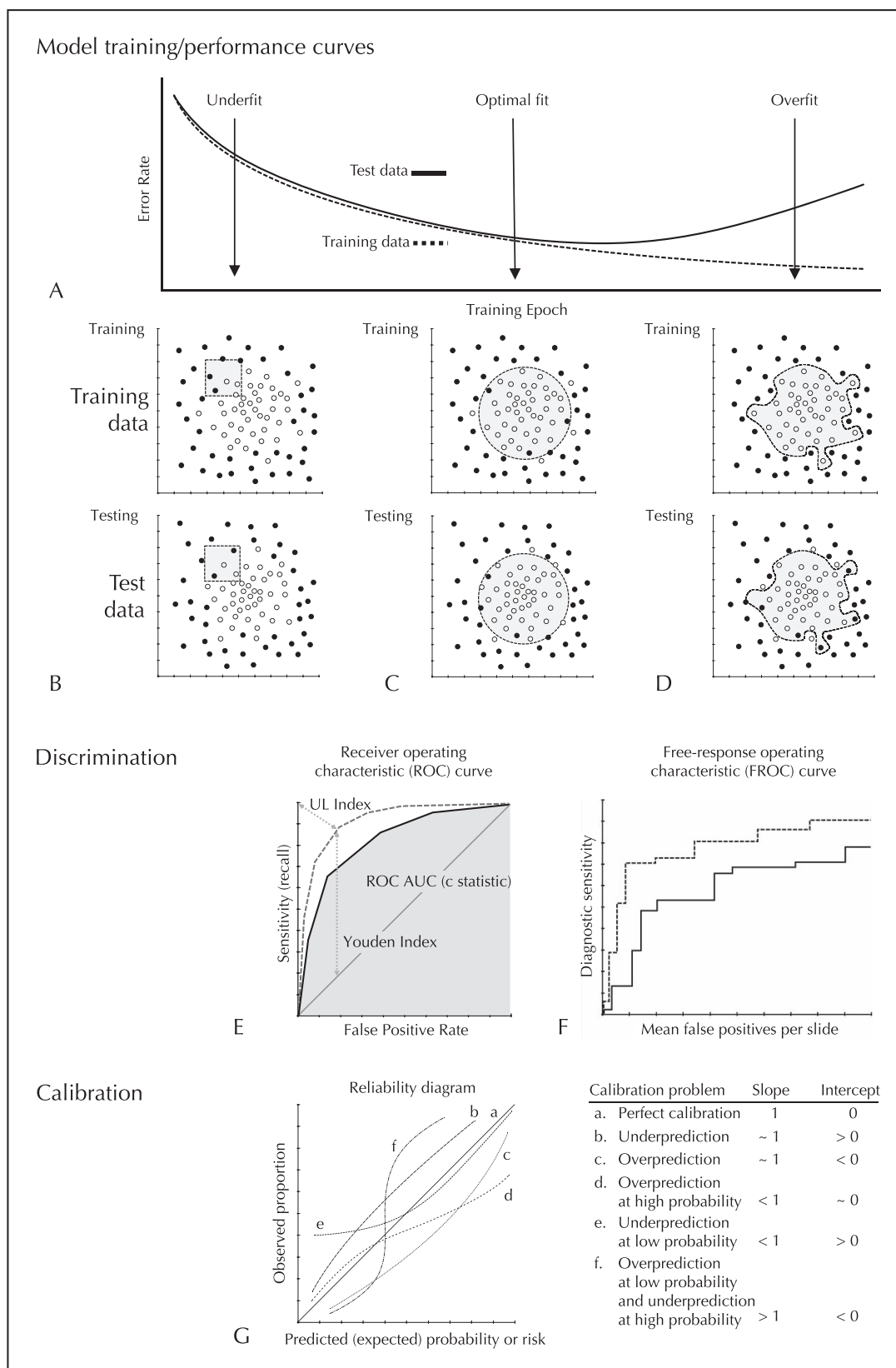


Figure 7. Evaluation of discrimination and calibration. Models are underfit early in training (A and B) and perform poorly on both training and test data because the boundaries for discriminating between the classes have not been optimized. In this illustration, the location, size, and shape of the boundary (the shaded square) is initially incorrect and the model is underfit. As training progresses, the location of the discrimination boundaries is optimized to reflect the underlying true data relationships (C, optimal fit), and performance on both the training and test data is good (A). If training continues, a model can begin to incorporate details specific to the training data that do not reflect the true underlying data relationships (the model is overfit to the training data, D). When this occurs, the performance against the training data can continue to improve, but the performance against the

data. A biased model is incorrect because it does not represent this relationship accurately, and it can be considered underfit. *Variance* refers to the sensitivity of a model to the details of a particular training set, such as noise or patterns in unimportant features, and a model with high variance can be considered overfit. There is a tension between bias and variance in model training. At the start of training models have high bias (they are underfit and do not accurately represent the data), and this bias is reduced as training progresses. After many training epochs the model begins to incorporate excessive detail from the training set and variance rises (ie, the model becomes overfit). Models with many parameters to learn, such as neural networks or tall decision trees, are prone to overfitting, especially with many passes through limited training data. Simplifying the model or increasing the amount of training data can help decrease overfitting and variance. In bootstrapping and cross-validation, the average performance across the individual data sets reflects bias, and the standard deviation between the sets reflects variance. The relationship between bias and variance can be seen in progressive performance curves during training (Figure 7, A). The goal of training is to reach the optimal bias/variance balance and not go beyond.

An additional form of bias occurs when the data used to train and test a model do not accurately reflect the intended deployment context. This problem may occur when the training set has systematic missing data or errors, when there is nonrandom collection of data for training, or when the training set may reflect outmoded practices or other differences from the deployment context. In these cases, the data relationships defined in the model during training do not accurately reflect the deployment context, and the performance of the model is lower than expected. In health care contexts, historically underserved minority groups are particularly susceptible to these types of subtle biases because their data in retrospective data sets may be underrepresented or distorted in other ways. Recently, this type of bias was discovered in a widely used algorithm that incorrectly reduced assignment of follow-up health care resources to minority patients by more than 60%.¹³⁴ Sex imbalances in training data may also produce inaccurate models.¹³⁵ Avoiding these problems requires careful consideration of training data collection methods, verification that the training data are appropriately representative of the deployment context, and verification of expected model performance at the deployment site.

Performance problems may also occur when classes of instances are unequally represented in a population.⁶⁷ Standard training methods assume that classes are relatively

balanced and use overall accuracy as a primary training metric by default. When there is imbalance in class incidence (ie, there is a minority class), a model trained to maximize overall accuracy will tend to overemphasize the majority class. If correct identification of the minority class is important, as is common in health care screening settings, the focus of training must be modified. Sensitivity or F1 score for the minority class, or metrics designed for class imbalance, such as the Matthews Correlation Coefficient (Table 1), may be substituted for overall accuracy as a primary metric for optimization during training, and errors in assignment of the minority class may be weighted more heavily than majority class errors. When training with unbalanced data sets, it is important to stratify cross-validation such that the folds contain representative and equivalent proportions of classes. In some cases, it may be useful to create artificially balanced training data by randomly undersampling the majority class or oversampling the minority class. Oversampling methods may randomly resample (duplicate) minority class instances or create new synthetic but plausible instances of the minority class (synthetic minority oversampling technique, SMOTE¹³⁶). ROC curves do not depict performance differences across changes in class incidence, so when that is important, other methods, such as precision-recall curves, may be warranted.^{137,138}

Many classifiers are binary and exclusive, that is, they support assignment to only 1 of 2 classes. Some classification tasks require assignment to 1 of more than 2 classes (multiclass problems), and some require the ability to assign more than 1 class to an instance (multilabel problems). These tasks may occur in a variety of machine learning settings, including image classification in pathology, where multiple features of interest or diagnoses may be present in a single slide image. Some algorithms can handle multiclass problems inherently, such as random forests and neural networks, and techniques are available to train ensembles of binary classifiers to support assignment among multiple classes. Some algorithms can also handle multilabel assignment. For example, k-nearest neighbors can allow assignment to all classes within a defined distance from a new instance. Analogous to multiclass methods, ensembles of classifiers may be trained to handle multilabel assignment. Assessing performance on multiclass and multilabel classification tasks can be complex, because both overall performance as well as class-, label-, and instance-specific classification performance must be considered.¹²²

Pathology image classifiers raise additional performance assessment challenges. Whole slide images are large and

test data declines (A). The discrimination capability of models is evaluated using receiver-operating characteristic (ROC) curves (E) that are created by plotting sensitivity versus false-positive rate as the threshold for class assignment is varied over the operating range of the model. The gray 45° line represents random assignment. Models with overall higher discrimination performance have curves that more closely approach the upper left of the graph (dashed line versus solid black line), with higher sensitivity at lower false-positive rates. The area under the ROC curve (ROC AUC, also called the C-statistic, shown in gray for the solid curve, E) is a measure of overall discrimination performance, with an AUC of 1 being perfectly correct, 0.5 being random, and 0 being perfectly incorrect. The Youden Index (greatest vertical distance between the model and the random line) and UL Index (shortest distance between the model and the upper left of the graph) indicate the point where the ratio of sensitivity to false-positive rate is maximal. Free response operating characteristic (FROC) curves (F) are used for evaluation of whole slide imaging performance and express sensitivity versus the average number of false positives per slide. Analogous to ROC, curves approaching the upper left are desirable (higher diagnostic sensitivity with lower false positives, dotted line). Because the number of false positives per slide is not fixed, FROC AUCs are not comparable in the same way as ROC AUCs. Calibration for a classifier is evaluated using reliability diagrams (G) that display model predictions in comparison with observed outcomes (observed/expected). The curves may represent average predictions and outcomes for centiles of the predicted probability, or probability functions for the predicted and observed values. Perfect calibration is represented by the diagonal solid line (a) showing equal predictions and outcomes across the range, with a slope of 1 and intercept of 0. Example calibration problems are shown in the dashed/dotted lines with linear slopes and intercepts listed, including general underprediction (b), general overprediction (c), overprediction at high probabilities (d), underprediction at low probabilities (e), and overprediction at low probability combined with underprediction at high probability (f).

Table 2. Preanalytic Sources of Image Variation in Digital Pathology^a

Tissue handling
Collection timing and temperature
Size of tissue
Fixative source, type, and storage
Fixation timing and conditions
Processing time and technique
Slide preparation
Section thickness
Section flatness
Tissue artifacts (crush, folds, knife marks)
Other artifacts (floaters, bubbles)
Stain variation
Staining timing and process
Reagent sources
Immunohistochemistry platform
Image acquisition
Magnification
Focus
Resolution
Scanner sensitivity
Data compression and file type

^a Data derived from primary sources.^{177,178}

may contain multiple diagnostic areas, any of which might be sufficient to assign a diagnosis for the slide. Because ROC curves assess only the final classification of an instance, they do not accurately depict the ability of a model to identify and classify correctly multiple regions of interest within a slide. To better represent this capability, some recent studies in pathology imaging have used FROC for assessing discrimination.^{139,140} The FROC curves plot sensitivity (recall) versus the mean number of false positives per image (Figure 7, F). The plots have superficial similarity to ROC curves in that better performance is indicated by curves approaching the upper left (high sensitivity with low false-positive rate), but there is no optimal AUC because the number of false positives per image does not have a defined upper bound.

MODEL DEPLOYMENT

Verified machine learning models are removed from the training environment and incorporated into software tools or medical devices for deployment (Figure 6, g and h). The deployed system must feed data to the model in the same form as the training and testing data and receive the model's regression or classification output for reporting or additional processing. Models that are embedded in devices that provide a standardized internal data and operating environment may be deployed across multiple sites based on the device developer's verification of the model for use in that device. Models that are developed locally using a site's operational data may be deployed at the development site based on the results of verification against the test data, as described above. Models that are intended for deployment across multiple sites using local operational data are a special case. Data such as locally prepared and scanned histologic slides and contents of local laboratory information systems or EHR are subject to site-specific variation (Table 2 shows preanalytic variables for digital pathology) that may differ in unpredictable ways from the model development

data. Models intended for use with this type of data should undergo additional site-specific verification of performance (Figure 6, h). Because operational data may change over time as medical best practices and local site processes evolve, models using site data should also undergo periodic reverification to make sure that local changes have not degraded performance. At this early stage of the development of the field, consensus on best practices for local verification and performance monitoring have not been established.

LOCAL PERFORMANCE VERIFICATION

The ability of machine learning models to perform well with different but related data, such as data from different sites or closely related tasks, is referred to as *generalizability*. A model that generalizes well shows good performance against data from the site providing its development data as well as equivalent data from other sites. A model that generalizes poorly shows good performance at the development site but lower performance at other sites. Models do not generalize across sites when there are mismatches between development data and site data. At a gross level, unavailability of data at a site may block use of a model unless collection of the missing data can be instituted. For example, a simple sepsis prediction model developed previously included the percentage of bands as one of the inputs. This model became unusable as sites implemented modern automated differentials without a bands count. Differences in data representation (syntax, units) and data quality (numbers and types of errors, noise) also may be problematic. More subtle differences in development and site data can also reduce generalizability, for example, differences in class balance (eg, case mix) and process or technical differences that change the distributions of values of features (eg, staining, cutting, and imaging differences; Table 2; and analytic or health care process differences in nonimaging settings). These data changes essentially render the model underfit (biased) in the new setting, similar to the incompletely trained model shown in Figure 7, A and B.

Although the ability to generalize is critical for broad deployment of machine learning tools, investigation of generalizability has been limited. In a survey of more than 500 studies of machine learning in medical imaging, 94% used data from 1 site only.¹⁴¹ Studies that have addressed multisite generalizability of statistical and machine learning models have documented discrimination and calibration problems based on data differences between sites,^{142–144} and there is consensus that model development should incorporate data from multiple sources.^{98,128,129} It may be possible to mitigate some generalizability problems by aggregating training and testing data from multiple sites, with appropriate weighting,¹⁴⁵ but such data aggregation does not guarantee optimal performance at the each contributing site (Figure 8, A).¹⁴³ Aggregated data that incorporate variation across sites may effectively be “noisier,” and as training data they may yield more consistency overall but lower performance at each site than training on site-specific data. Ultimately it may be possible to reduce site-specific variation of imaging data through process standardization and image normalization,¹⁴⁶ but the impact these techniques may have on generalizability is difficult to forecast. Methods are under investigation that may allow statistical comparison of development and site data to determine the

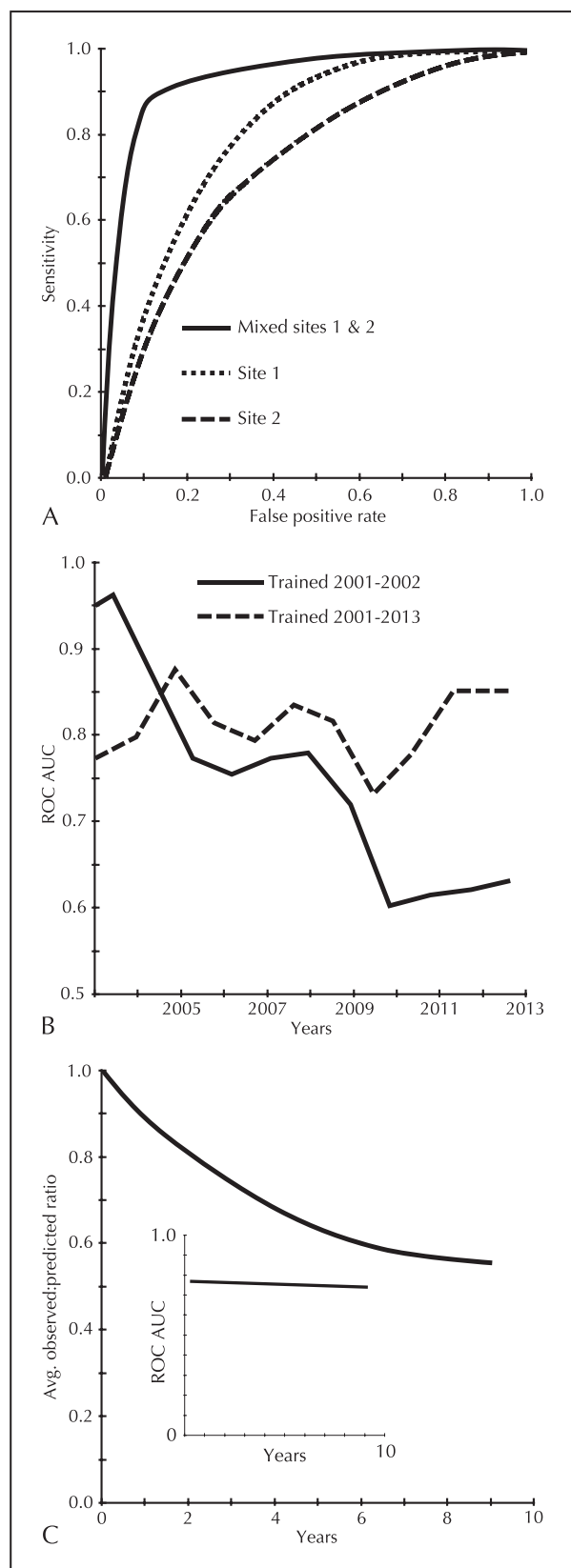


Figure 8. Machine learning model performance problems. Models may fail to generalize when data vary between training and deployment contexts (A), and creating an artificial mixture of data across sites for training may not solve this problem. These receiver-operating characteristic (ROC) curves show that a model for radiologic diagnosis trained on mixed data from 2 sites performed well against mixed test data (solid

applicability of models to particular sites or the need for corrective measures.¹⁴⁷

The potential for generalizability problems requires that sites verify model performance locally prior to deployment for routine use. Site-specific verification may be similar in form to developer verification of the model against the held-out test data, using expert-labeled site data (Figure 6, h). Performance metrics, such as sensitivity, specificity, predictive value, and calibration curves, may be calculated and compared to developer benchmarks. The goals for site performance verification should include detection and resolution of data mismatch problems, verification that discrimination and calibration performance meet expected benchmarks and local requirements, detection of locally relevant bias, and verification that the model is adequately robust, that is, the types and frequency of errors are manageable (Table 3). Appropriate fit with the local clinical workflow is also an important success factor for machine learning systems³⁰ that should be part of their initial planning, and workflow compatibility should be confirmed during deployment testing. If performance goals are not met by the native model, it may be possible to improve model performance by limited additional training on local data⁴⁴ (ie, "tuning" analogous to transfer learning discussed above) or by mathematically recalibrating the model.^{129,148} Local tuning and verification may require significant effort, depending on the required techniques and data volume, but early studies suggest that it is likely to be within the capabilities of most laboratories.⁴⁴

LOCAL PERFORMANCE MONITORING

Health care is a dynamic domain that continually incorporates new processes and procedures. These changes may be discrete and identifiable (eg, an EHR or laboratory information systems change, a Six Sigma project with process change, a new clinic or outreach client with a different case mix, or a change in a laboratory test with a reference range update), or they may result from an accumulation of small changes, such as minor policy or therapeutic changes in individual medical services, or gradual case mix changes. In the machine learning field these types of data dynamics are called covariate shifts (discrete changes in the distribution of input parameters) and data, or concept, drifts (gradual accumulated changes in input data¹⁴⁹). Shifts and drifts affecting input data for machine learning models may cause loss of discrimination and calibration (Figure 8, B and C). Loss of performance may be sudden or gradual depending on the dynamics of the data change,^{150–152} and it may involve loss of calibration

(line) but had lower performance against pure data from either site (dotted lines).¹⁴³ Changes in data over time (drifts, shifts) may reduce model performance (B). This example¹⁵² illustrates initially very good but later falling ROC area under the curve (AUC) for prediction of intensive care unit length of stay in males when a model was trained on electronic health record (EHR) data obtained prior to model use. In contrast, AUC was lower initially but remained stable when training data were taken from the entire time span of model use (2001–2013), and therefore incorporated temporal changes. Models may lose calibration (ability to predict probabilities accurately) without substantial changes in overall discrimination (C). In this example with a model that predicted acute kidney injury using EHR data,¹⁵³ the overall observed/predicted incidence of disease declined significantly over time, indicating that the model was overestimating the probability of disease, whereas the ROC AUC changed very little (inset).

Table 3. Goals, Needs, and Open Questions in Machine Learning Performance Assessment^a

Model development	
Goals	Create accurate, generalizable, robust models
Current options	Traditional training/testing pipelines
	Diversified data sources
	Operational-quality data reflecting real-world variation
Needs and open questions	Methods to determine optimal training/testing methods from operational performance requirements
	Methods to choose training and evaluation metrics that best represent the operational context
	Methods and metrics to evaluate model stability/robustness
Site verification	
Goals	Detect and correct data mismatch problems, confirm and document adequate model performance and stability
Current options	Verify by retesting discrimination and calibration using expert annotated local site data
	Stability with site data difficult to assess
Needs and open questions	Methods to measure relevant similarity and potential bias between developmental and local data sets
	Methods to define the optimal size and characteristics of local data sets for verification
	Methods to guarantee model stability within defined operational boundaries
	Methods to remediate generalizability problems
Performance monitoring	
Goals	Confirm and document adequate and stable model performance
Current options	Periodically report performance to the developer and the US Food and Drug Administration
	Periodic evaluation of discrimination and calibration using parallel analysis by experts
Needs and open questions	Methods to detect relevant data shifts and drifts rapidly
	Practical methods to measure and classify model performance without duplicate work
	Methods to remediate loss of discrimination or calibration

^a Data derived from Ashmore et al.¹⁶²

without a clear change in overall discrimination.¹⁵³ The potential (and expectation) for loss of performance over time necessitates ongoing performance monitoring at each deployment site after the initial system verification.

Future methods may be able to determine a particular model's susceptibility to input data change¹⁵⁴ so that ongoing monitoring of data can identify potentially important changes as they occur in, for example, feature distributions or class incidence.¹⁴⁷ Verification of discrimination and calibration may require periodic parallel interpretation of cases by experts or alternative systems, with selection of test cases across the class probability range to allow for calibration assessment. If discrimination or calibration declines below acceptable levels, it may be possible to mathematically recalibrate models to recover good performance without the need for full retraining.¹⁴⁸ If retraining is necessary and expert-labeled data are limiting, tuning by additional training on limited new site-specific data may be more effective than full retraining with less than optimal data volumes.¹⁴⁸

Goals for performance monitoring are listed in Table 3 and include verification of adequate performance on local data and reporting performance problems to the developer and the FDA. A performance monitoring program might also include interlaboratory performance comparisons analogous to current proficiency testing, but the relevant metric is likely to be the performance quality of each site's system on its own local data. Because of generalizability issues and local tuning, distributing common test images may not yield a result useful in verifying local performance unless methods are developed to tune the test images to each site's tissue processing and imaging characteristics. As with site verification, statistical methods, acceptable performance thresholds, and best practices for performance monitoring are currently investigational.

MODEL STABILITY

Humans understand realistic images and data to be depictions of the real world, and interpret them based on the real-world characteristics of the items or systems they represent. Although humans can be fooled and are subject to their own inconsistencies and biases, small changes to data that do not render the represented elements unrecognizable do not usually change the interpretation of the data. Larger changes that depict unreasonable relationships are recognized as unrealistic, and when errors occur they are usually between choices that have real-world similarities. Hence, human expert interpretation is relatively robust to small, extraneous changes in data. When artifacts or unrealistic elements make humans unsure about data interpretation, humans typically become more conservative, and they tend to avoid making risky choices when the cost of error is high. None of these considerations apply to machine learning models unless the capabilities are programmed in explicitly. Models operate purely on patterns derived from their training data without a view of the world beyond what is in those data. If a high probability of correctness is required before a decision with significant error-related consequences is made, that decision threshold must be implemented in the software. Because models do not incorporate a broad understanding of the world, they often do not err in the same way a human would. Small changes in data that would not change human interpretation but fall within gaps in the patterns of training data can yield unpredictable results. Models that may output significantly different results when input data vary only slightly are said to be unstable or "brittle."

Image classification models may show instabilities in which images that appear to humans to be very similar are classified differently. Searching for images that exploit these


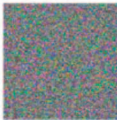


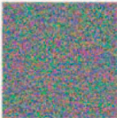

True label	Original	Adversarial Noise			Class probability	
		Noise	Modified		Original	Modified
Benign				Benign	89.6%	0.2%
				Malignant	10.4%	99.8%
Malignant				Benign	3.0%	74.6%
				Malignant	97.0%	25.4%

Figure 9. Model output stability. Images of benign and malignant skin lesions were classified correctly with a high probability by a convolutional deep neural network. Adversarial noise patterns were identified that, when added to the images, caused the network to misclassify them with high probability. The modified images are not easily distinguishable from the original by humans. Similar results were obtained with radiologic and retinal images (not shown). Images and data used with permission.¹⁵⁷

instabilities is termed an *adversarial attack* on a classifier, and the resulting images that provoke misclassification are called adversarial images. A number of methods are available for adversarial attacks.¹⁵⁵ The most widely used begin with correctly classified images and add small perturbations that progressively reduce the probability of class membership calculated by the model, manipulating the image to maximize the loss function of the model for that image until it is misclassified. Some methods then minimize the difference with the original image while preserving the misclassification. Effective perturbations may include rotation or mirroring of some or all of an image, or applying textures, noise, or masks to an image, and these changes may be imperceptible to humans. For example, classification of normal fundoscopic images and chest X-rays can be changed to diseased and diseased to normal, and classification of benign skin photographs can be changed to malignant and malignant to benign with minor image modifications not clearly perceptible to humans (Figure 9).^{156,157} Such images could be used in malicious attacks on systems, or images with adversarial characteristics might occur naturally and yield unpredictable classification errors. Adversarial examples also occur outside of the imaging context, suggesting that small, medically plausible perturbations in patient data might yield instability in personalized medicine recommendations or clinical trial group classifications.¹⁵⁸ However, adversarial examples may also become tools for addressing stability problems. They have been used to identify variation in stability among medical image classification models,¹⁵⁹ suggesting that in the future more robust models could be selected during model development. It may also become possible to use a paired network design analogous to GANs (see neural networks, above) to test model stability so that an acceptable classification failure rate can be guaranteed,¹⁶⁰ or to define a model of domain variation that supports robustness through comprehensive training.¹⁶¹

REGULATION OF ARTIFICIAL INTELLIGENCE

AI and especially machine learning algorithms introduce a fundamentally new kind of data analysis into the health care workflow. By virtue of their influence on pathologists and other physicians in selection of diagnoses and treatments, the outputs of these algorithms will critically impact patient care. To ensure that their use is safe and effective, the life cycles of tools based on these algorithms must be managed correctly. Models must be developed and tested in ways that limit harmful bias and instability, system performance must be verified as adequate for clinical purposes at installation,

the tools must be used correctly, and system performance must be monitored and reverified during use.¹⁶² These general requirements are familiar to clinical laboratories, but specific methods for accomplishing them in the context of machine learning are not well understood in the laboratory community. Creation of a regulatory framework with defined best practices for accomplishing these goals is a necessary step for successful dissemination of machine learning in pathology and medicine.

Effective and equitable regulations for AI in health care will (1) define requirements based on risk (ie, be tailored to the likelihood and magnitude of possible harm from each machine learning application), (2) require best practices for system development by vendors, including bias assessment and mitigation, (3) define best practices for verification of system performance at deployment sites, such as local laboratories, (4) define best practices for monitoring the performance of machine learning systems over time and mitigating performance problems that develop, (5) clearly assign responsibility for components of the verification and monitoring process, (6) clearly define reporting requirements for performance data, including performance problems, and (7) promote clarity and transparency in the use of machine learning tools that engenders trust from care providers and the general public. Regulatory strategy should encourage innovative development and improvement while meeting requirements for safety and efficacy, and therefore “least burdensome” approaches are favored.^{163,164} Risk-based strategies, in which regulatory requirements become more stringent as autonomy of operation and the potential magnitude of harm from error become greater, are part of a least burdensome approach.

Regulation of AI is in early development and is of international interest. In January 2020 the US White House released draft guidance that provides a set of high-level principles to which a regulatory framework for AI in any domain should adhere.¹⁶⁵ Key elements include public trust and participation in regulatory development, scientific integrity and transparency, focus on risk assessment and management, consideration of regulatory cost/benefit, regulatory flexibility consistent with rapid evolution and improvement of AI systems, ensuring fairness and nondiscrimination in AI operation, transparency in use of AI, safety and security in AI operation, and coordination between agencies for consistent approaches to regulation. The European Commission released a white paper¹⁶⁶ on AI in February 2020 that included an extensive discussion of AI regulation, including scope and technical requirements to ensure human oversight, robustness and safety, privacy and

appropriate data governance, transparency of operation, fairness and nondiscrimination, accountability, and promotion of societal well-being. These documents cover similar themes and provide high-level guidance for the development of regulations, but they do not provide detailed methods that would be useful for managing AI applications in an operational context.

Specific to health care in the United States, the FDA and the International Medical Device Regulators Forum have defined Software as a Medical Device (SaMD) as medical software that is not an integral part of a hardware device, and have created a framework for SaMD risk categorization, clinical evaluation, and quality management.¹⁶⁷ The FDA has also released a proposal for a streamlined pathway of software development leading to approval or clearance of SaMD based on precertification of software developers who have a strong track record of success and employ defined development best practices (the Pre-Cert program¹⁶⁸). Although AI software that is used as an independent tool to process laboratory data or histologic images would be classified as SaMD, these documents do not directly address the special verification and performance monitoring needs of AI. Recently the FDA also released a proposal for development pathways to update and improve AI software and to allow for AI SaMD that progressively learns.¹² The goal of this proposal is to clarify what constitutes a “new” device requiring review, and to reduce regulatory barriers to innovation and improvement in AI software. The proposal provides a life cycle strategy for AI software deployment and upgrade but refers back to the previous SaMD documents for the local verification and performance monitoring components of this life cycle. Hence, it does not address user needs for specific guidance and best practices for AI SaMD verification and management.

Traditionally the FDA has regulated the approval for sale of laboratory tests and testing devices, including the required content of package inserts that define methods for routine use of the test. The regulation of the laboratory service operation, including aspects of test verification, use of controls, linearity checks, user qualifications and training, and proficiency testing, is specified in the Clinical Laboratory Improvement Act (CLIA) regulations and enforced by the Centers for Medicare & Medicaid Services (CMS).¹⁶⁹ To date, CMS has not addressed regulation of the operational aspects of AI SaMD, and it is not clear whether or when CLIA regulations may apply to AI applications. There is potential for confusion, for example, regarding whether an AI application, such as an interpretation of the probability of sepsis given a cluster of laboratory test results, would fall under CLIA if implemented by laboratory personnel in the clinical laboratory, but not if implemented by hospital computing using laboratory data in the EHR. There are analogies between the features of some AI applications and regulated elements of complex clinical laboratory tests, for example, local verification, quality control for discrimination and calibration, monitoring for drifts and shifts, and user qualifications/training. It would be appropriate and helpful for these elements of AI system operation to be addressed in a comprehensive medical AI regulatory framework.^{164,170–173} However, AI SaMD monitoring is likely to require methodologies distinct from current test monitoring strategies and suited to its particular needs. As noted above, for example, meaningful proficiency testing for AI applications in imaging may require innovative methods to incorporate the specific characteristics of local tissue processing and

slide digitization into test material. Current regulations for the laboratory are written to address the needs of diagnostic testing, and although there are analogies with AI applications, these regulations do not map exactly to AI performance evaluation needs or provide useful guidance to local sites on AI performance monitoring methodology.

CONCLUSIONS

Exciting developments in machine learning, and particularly in complex multilayered neural networks (“deep learning”), have opened the door for the creation of new software capabilities and new tools with a high anticipated utility in medicine and pathology. Although there are as yet few commercial products in pathology based on these developments, results from research publications and early prototypes from startup and established companies are tantalizing. The form of the anticipated new tools is not yet clear, although there is evidence that best performance accrues from an optimal blend of human and machine capabilities—yielding “augmented intelligence” rather than artificial intelligence with the replacement of human expertise. Surveys indicate that pathologists are receptive to these new tools if they improve the quality of patient care. Pathologists also believe that they should retain final responsibility for clinical decision-making and management of the new tools, and that they will need additional education to assume the roles of expert user and knowledgeable manager.

Tools with the greatest impact in anatomic pathology are likely to be based on complex convolutional neural networks. Developing these types of models requires large amounts of data and high-performance computing for training, so they are likely to be built by academic consortia and commercial developers and deployed to clinical sites for use. There are many other types of machine learning algorithms that have different capabilities and different data requirements. These algorithms may be more appropriate than deep networks for particular applications in anatomic pathology, molecular diagnostics, and laboratory medicine, and in settings in which data volume is limiting. Future tools may employ ensembles of machine learning models, with different types of models dedicated to tasks for which they are appropriate. Hence, it is useful to be familiar with the range of machine learning algorithms and their typical applications.

Training and performance testing at development sites provide a performance benchmark for machine learning systems. Because of differences in data patterns between sites, the performance levels seen in development are not guaranteed to generalize to deployment sites. Most studies of machine learning performance in health care have used data from only a single site, so the understanding of potential generalizability issues is incomplete. Available evidence suggests that each deployment site will need to verify performance on its own data by comparing a trial of the model on an appropriate set of local data with the developer performance benchmark and with its own performance requirements. Studies also indicate that local data may change over time as local processes evolve, and the accumulation of these changes may reduce model performance. Ongoing performance monitoring will be necessary to address this potential problem and determine whether model performance remains adequate for local needs.

Pathologists should be responsible for verifying, monitoring, and determining clinical appropriateness of software deployed by the laboratory for clinical use, including machine learning software. There are clear analogies between machine learning models and the complex laboratory tests with which pathologists are familiar. However, critically evaluating machine learning software is challenging because a number of fundamental questions remain about methods for model development, deployment verification, and performance monitoring (Table 3). There is not currently a consensus on best practices to accomplish these tasks in a way that ensures patient safety and comprises due diligence, for either commercial or locally developed AI software. A regulatory framework that specifies best practices and clarifies responsibilities, similar to the combination of FDA approval and CLIA service requirements, would be useful but is not yet in place. Additional work that better defines generalizability limitations, statistical methods of performance verification and monitoring, and the statistical relationships between data characteristics and model performance is needed to define system management methods that can be incorporated into a best practice regulatory framework. In the meantime, it is prudent for pathologists to learn about the scope of application, strengths, and limitations of machine learning in preparation for a future that will incorporate powerful new tools and new management responsibilities.

The authors wish to acknowledge helpful comments and suggestions from Andrew Beck, MD, PhD; Christopher Garcia, MD; Eric Glassy, MD; and Ronald Jackups Jr, MD, PhD; and to thank Mary Kennedy and Kevin Schap for expert administrative support. The authors also wish to thank Samuel Finlayson for helpful comments and permission to use the images and data in Figure 9.

References

1. Artificial Intelligence: how knowledge is created, transferred, and used. AI Resource Center Web site. December 11, 2018. Updated October 23, 2020. <https://www.elsevier.com/connect/resource-center/artificial-intelligence>. Accessed September 10, 2020.
2. Dutton T. An overview of national AI strategies. *Medium*. June 28, 2018. <https://medium.com/politics-ai/an-overview-of-national-ai-strategies-2a70ec6edfd>. Accessed September 10, 2020.
3. Kenneth Research. Global artificial intelligence in the healthcare industry by offerings, by technology, and by application – global size analysis and market forecast 2019-2025. February 24, 2020. <https://www.kennethresearch.com/report-details/artificial-intelligence-in-the-healthcare-market/10078358>. Accessed October 28, 2020.
4. Blue Ridge Academic Health Group. Separating fact from fiction: recommendations for academic health centers on artificial and augmented intelligence. <http://whsc.emory.edu/blueridge/publications/archive/BlueRidge2018-2019.pdf>. Accessed September 10, 2020.
5. American Medical Association. Augmented intelligence in health care. <https://www.ama-assn.org/system/files/2019-01/augmented-intelligence-policy-report.pdf>. Accessed September 10, 2020.
6. Academy of Medical Royal Colleges. Artificial intelligence in healthcare. January 28, 2019. <http://www.aomrc.org.uk/reports-guidance/artificial-intelligence-in-healthcare/>. Accessed September 10, 2020.
7. Bresnick J. Early adopters question usefulness, maturity of AI in healthcare. October 18, 2017. <https://healthitanalytics.com/news/early-adopters-question-usefulness-maturity-of-ai-in-healthcare>. Accessed September 10, 2020.
8. Ross C, Swelitz I. IBM pitched its Watson supercomputer as a revolution in cancer care: it's nowhere close. September 5, 2017. <https://www.statnews.com/2017/09/05/watson-ibm-cancer/>. Accessed September 10, 2020.
9. Mitchell JR, Bilbily A, Geis R, et al. Canadian Association of Radiologists white paper on artificial intelligence in radiology. *Can Assoc Radiol J*. 2018;69(2):120–135.
10. American College of Radiology. ACR Data Science Institute Web site. 2020. <https://www.acrdsi.org>. Accessed September 10, 2020.
11. Gottlieb S. FDA's comprehensive effort to advance new innovations: Initiatives to modernize for innovation. *FDA Voices*. August 29, 2018. <https://www.fda.gov/NewsEvents/Newsroom/FDAVoices/ucm619119.htm>. Accessed September 10, 2020.
12. U.S. Food and Drug Administration. Proposed regulatory framework for modifications to artificial intelligence / machine learning (AI/ML)-based software as a medical device (SaMD). <https://www.fda.gov/files/medical%20devices/published/US-FDA-Artificial-Intelligence-and-Machine-Learning-Discussion-Paper.pdf>. Accessed December 26, 2020.
13. Topol EJ. High-performance medicine: the convergence of human and artificial intelligence. *Nat Med*. 2019;25(1):44–56.
14. Bond WF, Schwartz LM, Weaver KR, Levick D, Giuliano M, Graber ML. Differential diagnosis generators: an evaluation of currently available computer programs. *J Gen Intern Med*. 2012;27(2):213–219.
15. Rosso C. What caused the AI renaissance? *Psychology Today*. January 10, 2019. <https://www.psychologytoday.com/us/blog/the-future-brain/201901/what-caused-the-ai-renaissance>. Accessed September 10, 2020.
16. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ, eds. *Advances in Neural Information Processing Systems* 25. Red Hook, NY: Curran Associates Inc; 2012:1097–1105.
17. Gulshan V, Peng L, Coram M, et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA*. 2016;316(22):2402–2410.
18. Grzybowski A, Brona P, Lim G, et al. Artificial intelligence for diabetic retinopathy screening: a review. *Eye (Lond)*. 2020;34(3):451–460.
19. Esteve A, Kuprel B, Novoa RA et al. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*. 2017;542(7639):115–118.
20. McBee MP, Awan OA, Colucci AT, et al. Deep learning in radiology. *Acad Radiol*. 2018;25(11):1472–1480.
21. Acs B, Rantalainen M, Hartman J. Artificial intelligence as the next step towards precision pathology. *J Intern Med*. 2020;288(1):62–81.
22. Schinkel M, Paranjape K, Nannan Panday RS, Skyttberg N, Nanayakkara PWB. Clinical applications of artificial intelligence in sepsis: a narrative review. *Comput Biol Med*. 2019;115:103488.
23. Mlodzinski E, Stone DJ, Celi LA. Machine learning for pulmonary and critical care medicine: a narrative review. *Pulm Ther*. 2020;6(1):67–77.
24. Rashidi HH, Sen S, Palmieri TL, Blackmon T, Wajda J, Tran NK. Early recognition of burn- and trauma-related acute kidney injury: a pilot comparison of machine learning techniques. *Sci Rep*. 2020;10(1):205.
25. Li F, Liu W, Yu H. Extraction of information related to adverse drug events from electronic health record notes: design of an end-to-end model based on deep learning. *JMIR Med Inform*. 2018;6(4):e12159.
26. Shimabukuro DW, Barton CW, Feldman MD, Mataraso SJ, Das R. Effect of a machine learning-based severe sepsis prediction algorithm on patient survival and hospital length of stay: a randomised clinical trial. *BMJ Open Respir Res*. 2017;4(1):e000234.
27. Awan SE, Bennamoun M, Sohail F, Sanfilippo FM, Chow BJ, Dwivedi G. Feature selection and transformation by machine learning reduce variable numbers and improve prediction for heart failure readmission or death. *PLoS One*. 2019;14(6):e0218760.
28. Liang H, Tsui BY, Ni H, et al. Evaluation and accurate diagnoses of pediatric diseases using artificial intelligence. *Nat Med*. 2019;25(3):433–438.
29. Reddy S, Fox J, Purohit MP. Artificial intelligence-enabled healthcare delivery. *J R Soc Med*. 2019;112(1):22–28.
30. Beede E, Baylor E, Hersch F, et al. A human-centered evaluation of a deep learning system deployed in clinics for the detection of diabetic retinopathy. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. New York, NY: Association for Computing Machinery; 2020:1–12. doi: 10.1145/3313831.3376718
31. Fleishon HB, Haffty BG. Subject: (Docket No. FDA-2019-N-5592) “Public Workshop - Evolving Role of Artificial Intelligence in Radiological Imaging”: comments of the American College of Radiology. https://www.acr.org/-/media/ACR/NOINDEX/Advocacy/acr_rsna_comments_fda-ai-evolvingrole-ws_6-30-2020.pdf. Accessed September 10, 2020.
32. Wei JW, Tafe LJ, Linnik YA, Vaickus LJ, Tomita N, Hassanpour S. Pathologist-level classification of histologic patterns on resected lung adenocarcinoma slides with deep neural networks. *Sci Rep*. 2019;9(1):1–8.
33. Coudray N, Ocampo PS, Sakellaropoulos T, et al. Classification and mutation prediction from non-small cell lung cancer histopathology images using deep learning. *Nat Med*. 2018;24(10):1559–1567.
34. Yu KH, Zhang C, Berry GJ, et al. Predicting non-small cell lung cancer prognosis by fully automated microscopic pathology image features. *Nat Comm*. 2016;7:1–10.
35. Mobadersany P, Yousefi S, Amdam M, et al. Predicting cancer outcomes from histology and genomics using convolutional networks. *Proc Natl Acad Sci U S A*. 2018;115(13):E2970–E2979.
36. Korbar B, Olofson AM, Miralaf AP, et al. Deep learning for classification of colorectal polyps on whole-slide images. *J Pathol Inform*. 2017;8:30.
37. Achi HE, Belousova T, Chen L, et al. Automated diagnosis of lymphoma with digital pathology images using deep learning. *Ann Clin Lab Sci*. 2019;49(2):153–160.
38. Veta M, Heng YJ, Stathonikos N, et al. Predicting breast tumor proliferation from whole-slide images: the TUPAC16 challenge. *Med Image Anal*. 2019;54:111–121.
39. Liu Y, Gadepalli K, Norouzi M, et al. Detecting cancer metastases on gigapixel pathology images [posted online March 8, 2017]. *arXiv*. 2017;arXiv:1703.02442.

40. Liu Y, Kohlberger T, Norouzi M, et al. Artificial intelligence-based breast cancer nodal metastasis detection: insights into the black box for pathologists. *Arch Pathol Lab Med*. 2019;143(7):859–868.
41. Steiner DF, MacDonald R, Liu Y, et al. Impact of deep learning assistance on the histopathologic review of lymph nodes for metastatic breast cancer. *Am J Surg Pathol*. 2018;42(12):1636–1646.
42. Hasnain Z, Mason J, Gill K, et al. Machine learning models for predicting post-cystectomy recurrence and survival in bladder cancer patients. *PLoS One*. 2019;14(2):e0210976.
43. Nagpal K, Foote D, Liu Y, et al. Development and validation of a deep learning algorithm for improving Gleason scoring of prostate cancer. *NPJ Digit Med*. 2019;2(1):48.
44. Pantanowitz L, Quiroga-Garza GM, Bien L, et al. An artificial intelligence algorithm for prostate cancer diagnosis in whole slide images of core needle biopsies: a blinded clinical validation and deployment study. *Lancet Digit Health*. 2020;2(8):e407–e416.
45. Brinton LA, Fan S, Karssemeijer N, et al. Using deep convolutional neural networks to identify and classify tumor-associated stroma in diagnostic breast biopsies. *Mod Pathol*. 2018;31(10):1502–1512.
46. Bera K, Schalper KA, Rimm DL, Velcheti V, Madabhushi A. Artificial intelligence in digital pathology—new tools for diagnosis and precision oncology. *Nat Rev Clin Oncol*. 2019;16(11):703–715.
47. Bejnordi BE, Veta M, Van Diest PJ, et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *JAMA*. 2017;318(22):2199–2210.
48. Serag A, Ion-Margineanu A, Qureshi H, et al. Translational AI and deep learning in diagnostic pathology. *Front Med (Lausanne)*. 2019;6:185.
49. Fu Y, Jung AW, Torne RV, et al. Pan-cancer computational histopathology reveals mutations, tumor composition and prognosis. *Nat Cancer*. 2020;1:800–810.
50. Kather JN, Heij LR, Grabsch HI, et al. Pan-cancer image-based detection of clinically actionable genetic alterations. *Nat Cancer*. 2020;1:789–799.
51. Acs B, Hartman J. Next generation pathology: artificial intelligence enhances histopathology practice. *J Pathol*. 2020;250(1):7–8.
52. Colling R, Pitman H, Oien K, et al. Artificial intelligence in digital pathology: a roadmap to routine use in clinical practice. *J Pathol*. 2019;249(2):143–150.
53. Goldstein BA, Navar AM, Carter RE. Moving beyond regression techniques in cardiovascular risk prediction: applying machine learning to address analytic challenges. *Eur Heart J*. 2017;38(23):1805–1814.
54. Richardson A, Signor BM, Lidbury BA, Badrick T. Clinical chemistry in higher dimensions: machine-learning and enhanced prediction from routine clinical chemistry data. *Clin Biochem*. 2016;49(16–17):1213–1220.
55. Wilkes EH, Rumsby G, Woodward GM. Using machine learning to aid the interpretation of urine steroid profiles. *Clin Chem*. 2018;64(11):1586–1595.
56. Cabitza F, Banfi G. Machine learning in laboratory medicine: waiting for the flood. *Clin Chem Lab Med*. 2018;56(4):516–524.
57. Luo Y, Szolovits P, Dighe AS, Baron JM. Using machine learning to predict laboratory test results. *Am J Clin Pathol*. 2016;145(6):778–788.
58. Yu M, Bazydlo LAL, Bruns DE, Harrison JHJ. Streamlining quality review of mass spectrometry data in the clinical laboratory by use of machine learning. *Arch Pathol Lab Med*. 2019;143(8):990–998.
59. Liew C. The future of radiology augmented with Artificial Intelligence: a strategy for success. *Eur J Radiol*. 2018;102:152–156.
60. Lakhani P, Prater AB, Hutson RK, et al. Machine learning in radiology: applications beyond image interpretation. *J Am Coll Radiol*. 2018;15(2):350–359.
61. Chen PC, Gadepalli K, MacDonald R, et al. An augmented reality microscope with real-time artificial intelligence integration for cancer diagnosis. *Nat Med*. 2019;25(9):1453–1457.
62. Hegde N, Hipp JD, Liu Y, et al. Similar image search for histopathology: SMILY. *NPJ Digit Med*. 2019;2:1–9.
63. Kalra S, Tizhoosh HR, Shah S, et al. Pan-cancer diagnostic consensus through searching archival histopathology images using artificial intelligence. *NPJ Digit Med*. 2020;3:31.
64. Sarwar S, Dent A, Faust K, et al. Physician perspectives on integration of artificial intelligence into diagnostic pathology. *NPJ Digit Med*. 2019;2:28.
65. Wood MJ, Tenenholtz NA, Geis JR, Michalski MH, Andriole KP. The need for a machine learning curriculum for radiologists. *J Am Coll Radiol*. 2019;16(5):740–742.
66. Richter AN, Khoshgoftaar TM. A review of statistical and machine learning methods for modeling cancer risk using structured clinical data. *Artif Intell Med*. 2018;90:1–14.
67. Géron A. *Hands-On Machine Learning with Scikit-Learn & TensorFlow*. Sebastopol, CA: O'Reilly Media Inc; 2017.
68. Keogh E. Instance-based learning. In: Sammut C, Webb GI, eds. *Encyclopedia of Machine Learning*. Boston, MA: Springer; 2010.
69. Rashidi HH, Tran NK, Betts EV, Howell LP, Green R. Artificial intelligence and machine learning in pathology: the present landscape of supervised methods. *Acad Pathol*. 2019;6:2374289519873088.
70. Lindman K, Rose JF, Lindvall M, Lundström C, Treanor D. Annotations, ontologies, and whole slide images—development of an annotated ontology-driven whole slide image library of normal and abnormal human tissue. *J Pathol Inform*. 2019;10:22.
71. Lutnick B, Ginley B, Govind D, et al. An integrated iterative annotation technique for easing neural network training in medical image analysis. *Nat Mach Intell*. 2019;1(2):112–119.
72. Zhou ZH. A brief introduction to weakly supervised learning. *Nat Sci Rev*. 2018;5(1):44–53.
73. He J, Baxter SL, Xu J, Xu J, Zhou X, Zhang K. The practical implementation of artificial intelligence technologies in medicine. *Nat Med*. 2019;25(1):30–36.
74. Zhu X, Goldberg AB. Introduction to semi-supervised learning. In: Brachman RJ, Dietterich T, eds. *Synthesis Lectures on Artificial Intelligence and Machine Learning*. Williston, VT: Morgan & Claypool; 2009:9–19.
75. Campanella G, Silva VWK, Fuchs TJ. Terabyte-scale deep multiple instance learning for classification and localization in pathology [posted online September 27, 2018]. *arXiv*. 2018;arXiv:1805.06983.
76. Pan SJ, Yang Q. A survey on transfer learning. *IEEE Trans Knowl Data Eng*. 2010;22(10):1345–1359.
77. Alcorn MA, Li Q, Gong Z, et al. Strike (with) a pose: neural networks are easily fooled by strange poses of familiar objects. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Institute of Electrical and Electronics Engineers (IEEE); 2019:4845–4854. doi: 10.1109/CVPR.2019.00498
78. Tajbakhsh N, Shin JY, Gurudu SR, et al. Convolutional neural networks for medical image analysis: full training or fine tuning? *IEEE Trans Med Imaging*. 2016;35(5):1299–1312.
79. Gottesman O, Johansson F, Komorowski M, et al. Guidelines for reinforcement learning in healthcare. *Nat Med*. 2019;25(1):16–18.
80. Silver D, Huang A, Maddison CJ, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*. 2016;529(7587):484–489.
81. Bajracharya K. Reinforcement learning: Super Mario, AlphaGo and beyond. *Dimensionless*. October 1, 2018. <https://dimensionless.io/reinforcement-learning-super-mario-alpha-go/>. Accessed September 10, 2020.
82. Brownlee J. A tour of machine learning algorithms. 2019. <https://www.datasciencecentral.com/profiles/blogs/a-tour-of-machine-learning-algorithms>. Accessed September 10, 2020.
83. Lecun Y, Bengio Y, Hinton G. Deep learning. *Nature*. 2015;521(7553):436–444.
84. Wang S, Yang DM, Rong R, Zhan X, Xiao G. Pathology image analysis using segmentation deep learning algorithms. *Am J Pathol*. 2019;189(9):1686–1698.
85. Simonyan K, Vedaldi A, Zisserman A. Deep inside convolutional networks: visualising image classification models and saliency maps. Presented at The Second International Conference on Learning Representations; April 14–16, 2014; Banff, California. *arXiv:1312.6034v2*.
86. Nicholson C. A beginner's guide to LSTMs and recurrent neural networks. *AI Wiki. Pathmind*. <https://pathmind.com/wiki/lstm>. Accessed September 20, 2020.
87. Lipton ZC, Berkowitz J, Elkan C. A critical review of recurrent neural networks for sequence learning [posted online October 17, 2015]. *arXiv*. 2015;arXiv:1506.00019.
88. Jing B, Xie P, Xing E. On the automatic generation of medical imaging reports. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, vol. 1. Stroudsburg, PA: Association for Computational Linguistics; 2018:2577–2586. doi: 10.18653/v1/P18-1240
89. Malhotra P, Vig L, Shroff G, Agarwal P. Long short term memory networks for anomaly detection in time series. In: *2015 Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2015)*. Louvain-la-Neuve, Belgium: i6doc.com; 2015:89–94. <https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2015-56.pdf>. Accessed September 10, 2020.
90. Goodfellow IJ, Pouget-Abadie J, Mirza M, et al. Generative adversarial networks [posted online June 10, 2014]. *arXiv*. 2014;arXiv:1406.2661.
91. Yi X, Walia E, Babyn P. Generative adversarial network in medical imaging: a review. *Med Image Anal*. 2019;58:101552.
92. Hou L, Agarwal A, Samaras D, Kurc TM, Gupta RR, Saltz JH. Unsupervised histopathology image synthesis [posted online December 13, 2017]. *arXiv*. 2017;arXiv:1712.05021.
93. Kadurin A, Nikolenko S, Khrabrov K, Aliper A, Zhavoronkov A. druGAN: an advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico. *Mol Pharm*. 2017;14(9):3098–3104.
94. Anand N, Huang P. Generative modeling for protein structures. In: *Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*. San Diego, CA: Neural Information Processing Systems Foundation Inc.; 2018;32:7494–7505. <http://papers.nips.cc/paper/7978-generative-modeling-for-protein-structures.pdf>. Accessed September 10, 2020.
95. Westerlund M. The emergence of deepfake technology: a review. *Technol Innov Manage Rev*. 2019;9(11):40–53.
96. Fabiya SD. A review of unsupervised artificial neural networks with applications. *Int J Comput Appl*. 2019;181(40):22–26.
97. Klein ME, Dabbs DJ, Shuai Y, et al. Prediction of the Oncotype DX recurrence score: use of pathology-generated equations derived by linear regression analysis. *Mod Pathol*. 2013;26(5):658–664.
98. Shipe ME, Deppen SA, Farjah F, Grogan EL. Developing prediction models for clinical use using logistic regression: an overview. *J Thorac Dis*. 2019;11(suppl 4):S574–S584.

99. Bennett TD, Russell S, King J, et al. Accuracy of the epic sepsis prediction model in a regional health system [posted online February 19, 2019]. *arXiv*. 2019; arXiv:1902.07276.
100. Ranganathan P, Pramesh CS, Aggarwal R. Common pitfalls in statistical analysis: logistic regression. *Perspect Clin Res*. 2017;8(3):148–151.
101. Support vector machines: a guide for beginners. Quantstart Web site. <https://www.quantstart.com/articles/Support-Vector-Machines-A-Guide-for-Beginners>. Accessed September 10, 2020.
102. Yang J, Ding X, Zhu W. Improving the calling of non-invasive prenatal testing on 13-/18-/21-trisomy by support vector machine discrimination. *PLoS One*. 2018;13(12):e0207840.
103. He Y, Ma J, Ye X. A support vector machine classifier for the prediction of osteosarcoma metastasis with high accuracy. *Int J Mol Med*. 2017;40(5):1357–1364.
104. Harrison O. Machine learning basics with the k-nearest neighbors algorithm. *Towards Data Science*. Medium. September 10, 2018. <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761>. Accessed September 10, 2020.
105. Ali HR, Dariush A, Provenzano E, et al. Computational pathology of pre-treatment biopsies identifies lymphocyte density as a predictor of response to neoadjuvant chemotherapy in breast cancer. *Breast Cancer Res*. 2016;18(1):21.
106. Ali HR, Dariush A, Thomas J, et al. Lymphocyte density determined by computational pathology validated as a predictor of response to neoadjuvant chemotherapy in breast cancer: secondary analysis of the ARTemis trial. *Ann Oncol*. 2017;28(8):1832–1835.
107. Wu M, Zhong X, Peng Q, et al. Prediction of molecular subtypes of breast cancer using BI-RADS features based on a “white box” machine learning approach in a multi-modal imaging setting. *Eur J Radiol*. 2019;114:175–184.
108. Hendriks MP, Verbeek XAAM, van Vegchel T, et al. Transformation of the national breast cancer guideline into data-driven clinical decision trees. *JCO Clin Cancer Inform*. 2019;3:1–14.
109. McLachlan GJ, Bean RW, Ng SK. Clustering. In: Keith JM, ed. *Volume II: Structure, Function, and Applications*. New York: Springer Science+Business Media; 2017:345–362.
110. Dabbura I. K-means clustering: algorithm, applications, evaluation methods, and drawbacks. *Medium*. September 17, 2018. <https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a>. Accessed September 10, 2020.
111. Guo P, Banerjee K, Joe Stanley R, et al. Nuclei-based features for uterine cervical cancer histology image analysis with fusion-based classification. *IEEE J Biomed Health Inform*. 2016;20(6):1595–1607.
112. Combes C, Azema J. Clustering using principal component analysis applied to autonomy-disability of elderly people. *Decis Support Syst*. 2013;55(2): 578–586.
113. Bühlmann P. Bagging, Boosting and ensemble methods. In: Gentle J, Härdle W, Mori Y, eds. *Handbook of Computational Statistics*. Berlin: Springer; 2012:985–1022.
114. Mayr A, Hofner B, Waldmann E, Hepp T, Meyer S, Gefeller O. An update on statistical boosting in biomedicine. *Comput Math Methods Med*. 2017;2017: 6083072.
115. Descheppe M, Eeckloo K, Vogelaers D, Waegeman W. A hospital wide predictive model for unplanned readmission using hierarchical ICD data. *Comput Methods Programs Biomed*. 2019;173:177–183.
116. Vedomske MA, Brown DE, Harrison JH. Random forests on ubiquitous data for heart failure 30-day readmissions prediction. In: *12th International Conference on Machine Learning and Applications*. Institute of Electrical and Electronics Engineers; 2013:415–421. doi: 10.1109/icmla.2013.158
117. Bashir U, Kawa B, Siddique M, et al. Non-invasive classification of non-small cell lung cancer: a comparison between random forest models utilising radiomic and semantic features. *Br J Radiol*. 2019;92(1099):20190159.
118. Open Source Initiative. The Open Source definition. March 22, 2007. <https://opensource.org/osd>. Accessed September 10, 2020.
119. Chen PC, Liu Y, Peng L. How to develop machine learning models for healthcare. *Nat Mater*. 2019;18:410–414.
120. Moons KG, Kengne AP, Woodward M, et al. Risk prediction models, I: development, internal validation, and assessing the incremental value of a new (bio)marker. *Heart*. 2012;98(9):683–690.
121. Jovic A, Brkic K, Bogunovic N. A review of feature selection methods with applications. In: *38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. Institute of Electrical and Electronics Engineers; 2015:1200–1205. doi: 10.1109/MIPRO.2015.7160458
122. Wu XZ, Zhou ZH. A unified view of multi-label performance measures. In: *Proceedings of the 34th International Conference on Machine Learning*. New York, NY: Association for Computing Machinery; 2017;70:3780–3788. doi: 10.5555/3305890.3306072
123. Hossain M, Sulaiman MN. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*. 2015;5(2):1–11.
124. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: machine learning in Python. *J Mach Learn Res*. 2011;12:2825–2830.
125. Sokolova M, Lapalme G. A systematic analysis of performance measures for classification tasks. *Inf Process Manag*. 2009;45(4):427–437.
126. Bizzego A, Bussola N, Chierici M, et al. Evaluating reproducibility of AI algorithms in digital pathology with DAPPER. *PLoS Comput Biol*. 2019;15(3): e1006269.
127. Bissuel A. Hyper-parameter optimization algorithms: a short review. *Medium*. April 16, 2019. <https://medium.com/criteo-labs/hyper-parameter-optimization-algorithms-2fe447525903>. Accessed September 10, 2020.
128. Park SH, Han K. Methodologic guide for evaluating clinical performance and effect of artificial intelligence technology for medical diagnosis and prediction. *Radiology*. 2018;286(3):800–809.
129. Moons KG, Kengne AP, Grobbee DE, et al. Risk prediction models, II: external validation, model updating, and impact assessment. *Heart*. 2012;98(9): 691–698.
130. Van Calster B, McLernon DJ, van Smeden M, Wynants L, Steyerberg EW. Calibration: the Achilles heel of predictive analytics. *BMC Med*. 2019;17(1):230.
131. Van Calster B, Nieboer D, Vergouwe Y, De Cock B, Pencina MJ, Steyerberg EW. A calibration hierarchy for risk models was defined: from utopia to empirical data. *J Clin Epidemiol*. 2016;74:167–176.
132. Kumar A, Liang PS, Ma T. Verified uncertainty calibration. In: *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*. San Diego, CA: Neural Information Processing Systems Foundation Inc; 2019;33:3792–3803. <http://papers.nips.cc/paper/8635-verified-uncertainty-calibration.pdf>. Accessed October 21, 2020.
133. Hutter F, Kotthof L, Vanschoren J, eds. *Automated Machine Learning: Methods, Systems, Challenges*. New York, NY: Springer Nature; 2019. doi: 10.1007/978-3-030-05318-5
134. Obermeyer Z, Powers B, Vogeli C, Mullainathan S. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*. 2019; 366(6464):447–453.
135. Larrazabal AJ, Nieto N, Peterson V, Milone DH, Ferrante E. Gender imbalance in medical imaging datasets produces biased classifiers for computer-aided diagnosis. *Proc Natl Acad Sci U S A*. 2020;117(23):12592–12594.
136. Alghamdi M, Al-Mallah M, Keteyian S, Brawner C, Ehrman J, Sakr S. Predicting diabetes mellitus using SMOTE and ensemble machine learning approach: the Henry Ford Exercise Testing (FIT) project. *PLoS One*. 2017;12(7): e0179805.
137. Davis J, Goadrich M. The relationship between precision-recall and ROC curves. In: *Proceedings of the 23rd International Conference on Machine Learning (ICML '06)*. New York, NY: Association for Computing Machinery; 2006:233–240. doi: 10.1145/1143844.1143874
138. Saito T, Rehmsmeier M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS One*. 2015;10(3):e0118432.
139. Moskowitz CS. Using free-response receiver operating characteristic curves to assess the accuracy of machine diagnosis of cancer. *JAMA*. 2017; 318(22):2250–2251.
140. Chakraborty DP. A brief history of free-response receiver operating characteristic paradigm data analysis. *Acad Radiol*. 2013;20(7):915–919.
141. Kim DW, Jang HY, Kim KW, Shin Y, Park SH. Design characteristics of studies reporting the performance of artificial intelligence algorithms for diagnostic analysis of medical images: results from recently published papers. *Korean J Radiol*. 2019;20(3):405–410.
142. Isbell JM, Deppen S, Putnam JB, et al. Existing general population models inaccurately predict lung cancer risk in patients referred for surgical evaluation. *Ann Thorac Surg*. 2011;91(1):227–233.
143. Zech JR, Badgeley MA, Liu M, Costa AB, Titano JJ, Oermann EK. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS Med*. 2018;15(11):1–17.
144. Wang X, Liang G, Zhang Y, Blanton H, Bessinger Z, Jacobs N. Inconsistent performance of deep learning models on mammogram classification. *J Am Coll Radiol*. 2020;17(6):796–803.
145. Balachandhar N, Chang K, Kalpathy-Cramer J, Rubin DL. Accounting for data variability in multi-institutional distributed deep learning for medical imaging. *J Am Med Inform Assoc*. 2020;27(5):700–708.
146. Anand D, Ramakrishnan G, Sethi A. Fast GPU-enabled color normalization for digital pathology. In: *2019 International Conference on Systems, Signals and Image Processing (IWSSIP)*. Institute of Electrical and Electronics Engineers; 2019:219–224. doi: 10.1109/IWSSIP.2019.8787328
147. Goldenberg I, Webb GI. Survey of distance measures for quantifying concept drift and shift in numeric data. *Knowl Inf Syst*. 2019;60:591–615.
148. Davis SE, Greevy RA, Fonnesebeck C, Lasko TA, Walsh CG, Matheny ME. A nonparametric updating method to correct clinical prediction model drift. *J Am Med Inform Assoc*. 2019;26(12):1448–1457.
149. Žilobaitė I, Pechenizkiy M, Gama J. An overview of concept drift applications. In: Japkowicz N, Stefanowski J, eds. *Big Data Analysis: New Algorithms for a New Society*. Studies in Big Data. New York, NY: Springer International Publishing; 2016:91–114.
150. Chen JH, Alagappan M, Goldstein MK, Asch SM, Altman RB. Decaying relevance of clinical data towards future decisions in data-driven inpatient clinical order sets. *Int J Med Inform*. 2017;102:71–79.
151. Nestor B, McDermott MBA, Chauhan G, et al. Rethinking clinical prediction: why machine learning must consider year of care and feature aggregation. Presented at the Machine Learning for Health (ML4H) Workshop at NeurIPS 2018; December 8, 2018; Montreal, Canada. arXiv:1811.12583.
152. Nestor B, McDermott MBA, Boag W, et al. Feature robustness in non-stationary health records: caveats to deployable model performance in common

clinical machine learning tasks [posted online August 2, 2019]. *arXiv*. 2019;arXiv:1908.00690.

153. Davis SE, Lasko TA, Chen G, Siew ED, Matheny ME. Calibration drift in regression and machine learning models for acute kidney injury. *J Am Med Inform Assoc*. 2017;24(6):1052–1061.

154. Nelson K, Corbin G, Anania M, Kovacs M, Tobias J, Blowers M. Evaluating model drift in machine learning algorithms. In: *Proceedings of the 2015 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA 2015)*. Institute of Electrical and Electronics Engineers; 2015:162–169. doi: 10.1109/CISDA.2015.7208643

155. Ren K, Zheng T, Qin Z, Liu X. Adversarial attacks and defenses in deep learning. *Engineering*. 2020;6(3):346–360.

156. Finlayson SG, Bowers JD, Ito J, Zittrain JL, Beam AL, Kohane IS. Adversarial attacks on medical machine learning. *Science*. 2019;363(6433):1287–1289.

157. Finlayson SG, Chung HW, Kohane IS, Beam AL. Adversarial attacks against medical deep learning systems [posted online February 4, 2019]. *arXiv*. 2019;arXiv:1804.05296.

158. Papangelou K, Sechidis K, Weatherall J, Brown G. Toward an understanding of adversarial examples in clinical trials. In: Berlingerio M, Bonchi F, Gärtner T, Hurley N, Ilirim G, eds. *Mach Learn Knowl Discov Databases*. New York, NY: Springer International Publishing; 2019:35–51.

159. Paschali M, Conjeti S, Navarro F, Navab N. Generalizability vs. robustness: investigating medical imaging networks using adversarial examples. In: Frangi A, Schnabel J, Davatzikos C, Alberola-López C, Fichtinger G, eds. *Medical Image Computing and Computer Assisted Intervention*. New York, NY: Springer International Publishing; 2018:493–501.

160. Dvijotham K, Goyal S, Stanforth R, et al. Training verified learners with learned verifiers [posted online May 29, 2018]. *arXiv*. 2018;arXiv:1805.10265.

161. Robey A, Hassani H, Pappas GJ. Model-based robust deep learning [posted online May 20, 2020]. *arXiv*. 2020;arXiv:2005.10247.

162. Ashmore R, Calinescu R, Paterson C. Assuring the machine learning lifecycle: desiderata, methods, and challenges [posted online May 10, 2019]. *arXiv*. 2019;arXiv:1905.04223.

163. The least burdensome provisions: concept and principles. Docket number: FDA-2017. US Food and Drug Administration Web site. <https://www.fda.gov/regulatory-information/search-fda-guidance-documents/least-burdensome-provisions-concept-and-principles>. Accessed September 10, 2020.

164. Allen TC. Regulating artificial intelligence for a successful pathology future. *Arch Pathol Lab Med*. 2019;143(10):1175–1179.

165. Vought RT. Guidance for regulation of artificial intelligence applications. White House Web site. <https://www.whitehouse.gov/wp-content/uploads/2020/01/Draft-OMB-Memo-on-Regulation-of-AI-1-7-19.pdf>. Accessed September 10, 2020.

166. On artificial intelligence—a European approach to excellence and trust. European Commission Web site. https://ec.europa.eu/info/sites/info/files/commission-white-paper-artificial-intelligence-feb2020_en.pdf. Accessed September 10, 2020.

167. Software as a medical device (SaMD). US Food and Drug Administration Web site. <https://www.fda.gov/medical-devices/digital-health/software-medical-device-samd>. Accessed September 10, 2020.

168. Digital health software precertification (Pre-Cert) program. US Food and Drug Administration Web site. <https://www.fda.gov/medical-devices/digital-health/digital-health-software-precertification-pre-cert-program>. Accessed September 10, 2020.

169. Clinical Laboratory Improvement Amendments (CLIA). US Food and Drug Administration Web site. <https://www.fda.gov/medical-devices/ivd-regulatory-assistance/clinical-laboratory-improvement-amendments-clia>. Accessed September 10, 2020.

170. Babic B, Gerke S, Evgeniou T, Cohen IG. Algorithms on regulatory lockdown in medicine. *Science*. 2019;366(6470):1202–1204.

171. Reddy S, Allan S, Coghlan S, Cooper P. A governance model for the application of AI in health care. *J Am Med Inform Assoc*. 2020;27(3):491–497.

172. Parikh RB, Obermeyer Z, Navathe AS. Regulation of predictive analytics in medicine. *Science*. 2019;363(6429):810–812.

173. Gerke S, Babic B, Evgeniou T, Cohen IG. The need for a system view to regulate artificial intelligence/machine learning-based software as medical device. *NPJ Digit Med*. 2020;3:53.

174. Steyerberg EW, Vickers AJ, Cook NR, et al. Assessing the performance of prediction models: a framework for traditional and novel measures. *Epidemiology*. 2010;21(1):128–138.

175. Scikit-Learn Developers. Clustering performance evaluation. *Scikit-Learn Users Guide*. <https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation>. Accessed September 10, 2020.

176. Scikit-Learn Developers. Metrics and scoring: Quantifying the quality of predictions. *Scikit-Learn Users Guide*. https://scikit-learn.org/stable/modules/model_evaluation.html. Accessed September 10, 2020.

177. Hamilton PW, Bankhead P, Wang Y, et al. Digital pathology and image analysis in tissue biomarker research. *Methods*. 2014;70:59–73.

178. Jones AD, Graff JP, Darrow M, et al. Impact of pre-analytical variables on deep learning accuracy in histopathology. *Histopathology*. 2019;75:39–53.