Artificial Intelligence in Healthcare

A Technical Introduction

September 2019

This white paper is the product of a multidisciplinary team with collaboration across different areas and subject matter experts within AI in healthcare.

The AAIH team of authors and contributors:

- Brandon Allgood – CTO & Co-founder of Numerate
- Oscar Rodriguez – Chief Architect at BlackThorn Therapeutics
- Jeroen Bédorf – Senior System Architect at minds.ai
- Pierre-Alexandre Fournier – CEO at Hexoskin
- Artur Kadurin – CEO at Insilico Taiwan
- Alex Zhavoronkov – Founder and CEO at Insilico Medicine
- Stephen MacKinnon – VP of Research and Development at Cyclica
- Rafael Rosengarten – Founder and CEO at Genialis
- Michael Kremliovsky – Director of Medical Devices & eHealth at Bayer
- Aaron Chang – Strategy and Technical Advisor at AAIH
- Annastasiah Mudiwa Mhaka – Co-founder and Convener at AAIH

Problem Statement

The term Artificial Intelligence (AI) has become pervasive in conversations about the future of healthcare. AI has the potential to transform medicine through novel models of scientific discovery and healthcare delivery leading to improved individual and public health outcomes. Yet misunderstanding and miscommunication abound. Therefore, the concepts related to AI need to be defined and explained in order to elevate our general level of understanding of and discourse around the topic.

Purpose of the Document

- To introduce, define and clarify foundational topics, terms and concepts in AI with an emphasis on applications in healthcare, spanning the continuum from biomedical discovery, clinical development and patient care.
- To coalesce the Alliance for AI in Healthcare (AAIH), the community it serves, and collaborators around a common language.
- To provide a platform for follow-on activities, including whitepapers, in which AAIH and collaborators will engage.
- To serve as a reference, or lexicon, for future discussions and publications.

Target Audience

This whitepaper is intended for the broader healthcare community, including the scientific press, researchers, developers, and other technically inclined healthcare practitioners and administrators.

TABLE OF CONTENTS

INTRODUCTION

Artificial Intelligence (AI) has been studied by computer scientists for more than 70 years. The term 'Artificial Intelligence' itself was coined by John McCarthy in 1956 at the first workshop on the subject at Dartmouth College.[1] But the theory and topics that became known as AI have a much longer history.[2] Even so, it remains one of the most complex and misunderstood topics in computer science because of the vast number of techniques employed and the often-nebulous goals being pursued.

AI and healthcare have been bound together for over half a century. The DENDRAL project, an early Expert System based on AI techniques from Stanford in the 1960s, aimed at hypothesis formation and discovery in science. The primary focus was to determine organic compound structures by analyzing their mass spectra.[3] A lot of new theoretical and program language work was undertaken to make this possible. It was followed by MYCIN in the 1970s with the goal of identifying infection-causing bacteria and to recommend antibiotics, with dosage adjusted for the patient's weight. The concepts behind MYCIN were then generalized to all internal medicine in the 1980s with the CADUSEUS system, described at the time as the "most knowledge-intensive expert system in existence."[4] Also in the 1980s, several techniques were developed for use in drug discovery. Since then, the number of techniques and uses in healthcare has grown steadily against a wider backdrop of AI "summers and winters" (The summer/winter metaphor has been the comparison of choice for describing the cyclical rise and fall of interest in AI and expectancy/hype around its deliverables). Today we are experiencing unprecedented AI summer that many believe is an integral part of the Fourth Industrial Revolution.

While a comprehensive history of AI in healthcare is beyond the scope of this paper, these historical examples demonstrate that AI has already had an impact on healthcare. Almost all major healthcare organizations and life science companies are currently employing or investigating use of applications based on various AI technologies. The current success (and hype) of AI is driven largely by the increase in computing power, availability of cheap storage and fast networking, advancement of algorithms, and increased awareness due to highly visible and successful consumer use cases. However, navigating the growing interest in and buzz around such a large and nebulous subject demonstrates the need for a well-defined set of foundational concepts and terms.

Currently, the areas where AI has made the most advances are those possessing a large amount of structured data, where the problem to solve is well understood or straightforward to define (image recognition, language translation, etc.). The opposite is true for most

---

1 Moor, J., The Dartmouth College Artificial Intelligence Conference: The Next Fifty Years. *AI Magazine,* **27** (4), pp. 87-91 (2006)

2 Nilsson, N. J., *The Quest for Artificial Intelligence*, Cambridge: Cambridge University Press (2010)

3 Lindsay, R. K., et al., DENDRAL: a case study of the first expert system for scientific hypothesis formation. *Artificial Intelligence,* **61**, pp. 209-261 (1993)

4 Feigenbaum E. A. and McCorduck, P., *The Fifth Generation: Artificial Intelligence and Japan's Computer Challenge to the World*, Addison-Wesley (1983)

cases in healthcare. Data are generally hard to obtain because they are expensive, restricted, and often incomplete or fractured amongst different stakeholders. Commonly these data are complex, inherently high-dimensional, semi-structured or unstructured, and the questions to answer are not simple to frame. Nonetheless, decisions based on models in healthcare can dramatically impact the wellness and lives of patients. For these reasons, implementation, validation, and deployment of AI in healthcare requires detailed attention to safety and efficacy compared to much lower risk applications used in consumer products and services. Therefore, it will take time and effort to get it right. This primer should serve that goal by orienting the newcomer in the current understanding and development efforts for applications of AI in Healthcare.

DEFINITIONS

Intelligence

Intelligence comes from the Latin word *intelligere*, meaning 'understand.' Merriam-Webster defines intelligence as both "the ability to learn or understand or to deal with new or trying situations" and "the ability to apply knowledge to manipulate one's environment or to think abstractly as measured by objective criteria (such as tests)."[5] These definitions, while useful, are by no means agreed upon by researchers. In fact, Shane Legg and Marcus Hunter (both AI researchers) assembled over 70 definitions of intelligence from various fields.[6] More recently Max Tegmark, in his book *Life 3.0*, put forth a simple definition. He defines intelligence as having the **"ability to accomplish complex goals."**[7] Due to its breadth and simplicity, it serves as a good base definition.

Intelligent Agent

While there exist several variants for the definition of an Intelligent Agent[8], they all share the same foundation. An Intelligent Agent (also known as a Rational Agent[9]) is **an autonomous entity that directs its activities toward accomplishing complex goals by making observations of its environment through sensors, processing the inputs, and acting on the environment through actuators (or effectors).** Examples of Intelligent Agents are humans, dogs, thermostats, modern airplanes, etc. The focus of this work is on Artificial Intelligent Agents, referred to herein simply as Intelligent Agents. Notably, the concept of the Intelligent Agent

5 Merriam-Webster, Definition of Intelligence. https://www.merriam-webster.com/dictionary/intelligence.

6 Legg , S. and Hunter, M., A Collection of Definitions of Intelligence. (Oct 2006) http://www.vetta.org/documents/A-Collection-of-Definitions-of-Intelligence.pdf

7 Tegmark, M., *Life 3.0*, Knopf (2017)

8 Franklin, S. & Graesser, A., Is It an agent, or just a program?: A taxonomy for autonomous agents, Intelligent Agents III Agent Theories, Architectures, and Languages. ATAL (1996)

9 Russell, S. & Norvig, P, *Artificial Intelligence: A Modern Approach*, 3rd Ed., Prentice Hall Press

allows us to stay above the implementation details when discussing applications, but still account for the key elements necessary for learning, decision making, and acting. Intelligent Agents may lack certain elements (such as Software-Only Agents considered separately from underlying hardware) and be different in their degree of autonomy. Fully autonomous Intelligent Agent is synonymous to the concept of "General AI." This paper is focused on the technologies underlying development of software-only agents. The software-only agents lead the current explosion in the field of business applications. There are, however, many examples of cyber-physical systems, instruments controlled by computer-based algorithms, within Healthcare, including medical devices of various sorts. A future work will focus more specifically on the state of cyber-physical systems development and application in Healthcare when a significant degree of decision autonomy is envisioned.

More generally, the term Intelligent System helps to account for multiple agents working as a system or being loosely integrated by linking several hardware and software components.

Artificial Intelligence (AI)

Merriam-Webster defines Artificial Intelligence as "the capability of a machine to imitate intelligent human behavior."[10] This definition is problematic for a number of reasons, most of all being the comparison with "human behavior." This makes it too narrow in its view of intelligence. Intelligence is not exclusively human and many currently developed Intelligent Agents perform beyond the ability of human experts, albeit on narrow tasks only.[11] In his book, Max Tegmark refers to AI in the abstract as anything that is "non-biological [and has the] ability to accomplish complex goals." While concise, this definition is also problematic because there has been a great deal of recent work, both theoretical and practical, on biologically-based Intelligent Agents (for example, DNA-based Agent[12]). The definition of AI that comes from computer science is **the study of artificial intelligent agents and systems, exhibiting the ability to accomplish complex goals.** This definition is the most relevant to the current context. It is useful to further define two sub-classifications: General AI and Narrow AI.

General and Narrow AI

General AI, often referred to as Artificial General Intelligence (AGI), is **the exhibition of a full range of cognitive abilities or general intelligence actions by an intelligent agent or system.** An Intelligent System demonstrating AGI can learn, understand, or deal with novel input on an effectively infinite set of unrelated tasks. The term AGI has a history going back to 1997 but was popularized in the early 2000s when it was used to mean *human-level* artificial intelligence.[13] The issue with AGI being defined as human-level is the ambiguity. Humans differ in their intelligence levels and the use of humans as a measuring stick forces one to then define sub-human AGI and superhuman AGI. Many believe that an Intelligent

10 Merriam-Webster, Definition of Artificial Intelligence. https://www.merriam-webster.com/dictionary/artificial%20intelligence

11 Goertzel, B., Who coined the term "AGI"? (Aug 2011) http://goertzel.org/who-coined-the-term-agi/

12 Qian, L. et al., Neural network computation with DNA strand displacement cascades. *Nature,* **475**, pp. 368-372 (2011)

13 Goertzel, B., Ibid.

System demonstrating AGI, even a weak AGI, by its nature and ability to ingest knowledge, will inevitably become superhuman.[14] In the context of this work, AI will not include the concept of AGI. A deeper discussion of AGI is out of scope for this work.

Narrow AI, sometimes called Weak AI, is **the exhibition of the ability to learn, understand, or deal with novel input in a limited or pre-defined scope, most often a single task or a set of highly related tasks, by an intelligent agent or system.** The term Narrow AI is preferred to Weak AI, as it better communicates its nature, give that Intelligent Systems can exhibit very high performance on single tasks. A particular area in healthcare where Intelligent Software Systems show high performance is in the field of radiology. Recently, Google AI along with a number of medical research hospitals published a study in which an Intelligent Software System was able to outperform six radiologists in diagnosing lung cancer from low-dose computed tomography (CT) screening images when no prior CT images were available. The Application had an 11% reduction in false positives (images predicted to contain a tumor, but were wrong) and a 5% reduction in false negatives (images predicted to not contain a tumor, but were wrong).[15] For all current and future uses of the term AI, Narrow AI is inferred, unless otherwise specified.

Fields of study within AI

The field of AI study encompasses a number of different sub-fields. Within each sub-field, particular algorithms and methods have been developed to address key elements of Intelligent Agents: Sensing, Pattern Recognition, Knowledge Representation, Reasoning, Optimization, and Control. Much of AI research is focused on developing these methods to impart Intelligent Agents with improved performance, safety, and autonomy. One may categorize AI algorithms in various ways: historically, by behavior, by application area, etc. Here, two sub-fields are examined: one based on deductive reasoning, Symbolic AI, and one based on inductive reasoning, Machine Learning.

Symbolic AI

Symbolic AI is a collection of all methods in AI that are based on high-level "symbolic" (human-readable) representations of problems, logic and search. Symbolic reasoning was one of the earliest focuses of AI research, and it led to the emergence of the sub-field of Expert Systems. Expert Systems, introduced by Edward Feigenbaum at Stanford, are systems designed to solve problems by searching through large databases of knowledge using heuristic rules designed by experts. The DENDRAL system mentioned in the introduction was the first expert system commercialized by Feigenbaum.

Symbolic representations play important roles in abstracting knowledge, formal reasoning, providing human-machine interfaces, and for explainability and interpretability of Intelligent Agents' decisions by human operators.

---

14 Tegmark, M., *Life 3.0*, Knopf (2017)

15 Ardila, D., et al., End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. *Nature Medicine,* **25**, pp. 954–961 (2019)

MACHINE LEARNING (ML)

Machine learning is the study of algorithms and statistical models that computer systems use to perform specific tasks without using explicit instructions, relying on patterns and inference found in the training data and in the environment. All machine learning systems consist of training sets of data, learning algorithms, and a resulting model representation. The algorithm extracts patterns from the **training dataset** and produces a model that encodes those patterns in such a way that the model can be used to evaluate a new set of data, referred to as the **test dataset.** Typically, the model is simply a mathematical function.

In order to produce a machine learning model, a representation of the data must be selected. Each parameter in the representation should relate to the underlying phenomenon being modeled in a meaningful way when possible. For example, for a model of cancer risk, a set of meaningful parameters might include age, smoker/non-smoker, weight, number of alcoholic drinks per week, home address, etc. Parameters that are not related to cancer risk would be things like color of car or favorite music style. The inclusion of parameters unrelated to the phenomenon can be problematic because machine learning algorithms look for correlations. It is statistically likely that an unrelated parameter will contain a spurious correlation given that training sets are finite in size. This can lead to biases and false conclusions. In many cases the optimal representation is difficult to determine, therefore a significant effort should go into developing the representation before applying a machine learning algorithm.

Role of Data

ML infers its models from the Data, hence the Data is the fuel on which the ML engines run. As such, the quality, volume, and composition of the data are critical. Higher quality data leads to a better model, in most cases. The same goes for the volume. But quality and volume are often competing factors. Lowering the quality standard can often lead to a higher requirement for the amount of data. The right choice in this trade-off is problem dependent and will determine which ML algorithm will produce the superior model or insight.

As with the quality/quantity trade-off there is often a quantity/composition trade-off. An oversimplified example of this would be: if a model were trained on patient data representing 80% European, 10% African, and 10% Asian descent, yet the underlying population that the model would be applied to represented 50% European, 25% African, and 25% Asian descent. In such a case, the amount of data from patients of European descent in the training set may be reduced prior to building a model in an attempt to remove genealogical bias. What is more troublesome are biases that we don't know exist. This is one of the most difficult tasks when building a ML model.

Solving the Relevant Problem

Defining the problem to be solved for is a crucial first step towards building a ML model. This comes before assessing the types and quality of data available to train a model. How a model will be used is something that needs to be examined before building a model. This relates

to the Bias Section, below, but also relates to the question of how the model will be used to support decision-making. Will the model be treated as a binary decision tool? If so, what is the tolerance for false positives and false negatives? In the case of skin cancer diagnosis, a high false positive rate will result in unneeded trips to the hospital and a high false negative rate will lead to poor health outcomes. Does the model need to be a regression model? Scientists often want a quantitative output instead of a categorical output, but high-quality regression models have higher data requirements than binary or categorical models.

Model building frequency is another consideration. How often does new data arrive? How often will the model need to be retrained: monthly, weekly or continuously? Different modeling approaches and ML algorithms are more amenable to continuous (also known as on-line) or frequent retraining. On the opposite end of the frequency of model updates are medical devices with requirements to conduct risk analysis and, depending on the outcomes, some form of validation up to a full new regulatory submission.

Finally, how much is known about the underlying process that is being modeled? This is important in helping to determine how to represent the underlying training set to the Machine Learning algorithm. Machine Learning is concerned with making predictions based on a training set, therefore all correlations of variables in the training set with the labels will be picked up by the ML algorithm. The algorithm will not be able to distinguish between causation and correlation, so to improve the generalizability and trustworthiness of the resultant model, non-relevant correlations need to be removed in the feature sets before model building. This is a particular issue in Healthcare because the underlying causal structure of most biological processes often are not well understood.

Bias

Bias comes from a ML model containing erroneous assumptions. The erroneous assumptions come for the relationship between the training data and the test data, from an inappropriate choice of features to represent the data, or from the machine learning process itself. These aspects when properly orchestrated can also help to compensate for issues in the other. Bias can often be context specific. Certain biases can exist when applied to one test set, but not another. Bias is unavoidable given finite training sets, therefore metadata about how a ML model was trained, what data it was trained on, what method was used, etc. needs to accompany the model to inform the application to help avoid bias.

When the bias comes from the ML process itself it can arise from using the wrong ML method given the data, the representation and the application. It can also arise from overfitting to the training data. Overfitting is the tendency for a ML model to memorize the details of the training set rather than learn generalizable patterns. Put another way, it is a tendency for the ML algorithm to select an overly complex model given the problem. A technique called regularization is often employed to prevent overfitting. The proper choice of featurization can also help to prevent overfitting. A more technical discussion of overfitting and how it relates to bias is in the Appendix.

Inappropriate featurization of the data can also lead to bias. If the featurization is too complex in comparison to the training set size the higher the likelihood one or more of the features will be erroneously correlated with the phenomenon being modeled. If the representation of the training data used contains features that are not related to the phenomenon being modeled they may also lead to bias. There may be a correlation between the feature and the phenomenon being modeled which will lead to the ML algorithm inappropriately incorporating this feature into the model. The issue of choosing the right representation and how to avoid bias based on features are addressed below in the Representation Section and the Fairness Section, respectively.

Another source of biases in ML models comes from the relationship of the training data to the test data. One of the base tenants of ML is that the training set is a good representation of the test set (discussed in more detail in the Appendix). More technically stated, the background distribution of properties of the training data most be the same as that of the test data otherwise the model will develop a systematic bias. The issue here is that the training set is finite and will therefore never be a complete representation of the test set and data scientists do not often have the ability to dictate what data is available for modeling. This does not however mean that there is no way forward. Identification of the bias in the training set is the first step. This can often then be compensated for by the proper choice of representation and machine learning method.

Finally, the most fundamental (and benign) source of bias comes from the fact that when building a ML model a choice is made as to the parameter to optimize. This can lead to intrinsic bias in the model. For example, if there are two groups of patients, one tolerating pain and one which does not, it is impossible, in general case, to use a single regimen of any pain medication with significant side effects for the entire population without bias. One can optimize the algorithm for the group or for the entire population, but not for both.

Any avoidable (or intrinsic) bias should be disclosed and, ideally, become a part of the model characterization and qualification.

Types of Machine Learning

Supervised Learning

Supervised learning is a type of ML that is trained on a set of labeled data. Supervised learning algorithms generate predictive models, based on patterns detected in the training data features that correlate with the training data labels. The generated model, which is a mathematical function of the features, can then be used to predict the labels for unlabeled data. An example of labeled data is a set of molecules and measurements in a biochemical assay; the measurement being the label for each molecule.

Supervised machine learning algorithms generate the final model by searching for a function over the training set features that minimizes a loss function (sometimes called a cost function). Following from the above example, features for a molecule may be such things as

molecular weight, number of nitrogen atoms, number of oxygen atoms, etc. A simple loss function for a classification model might be the number of molecules predicted incorrectly active or incorrectly inactive in a biochemical assay. In this case the algorithm would construct a function of the features that both positively and negatively correlate with activity, such that the number of molecules in the training set that are incorrectly predicted are minimized. Once that function is determined it becomes the output model that can be used to predict activity for unlabeled molecules. For more detail, please see Appendix I: Deeper Technical Details.

The three broad types of supervised machine learning model functions are classification, ranking, and regression. Classification is used to learn and predict a categorical label, e.g. tumor detection. Ranking is used to learn and predict a relative ordering label, e.g. diagnosis ranking. Regression is used to learn and predict a continuous label, e.g. assay readout prediction.

Unsupervised Learning

Unsupervised learning is a type of ML used to find (sometimes hidden) patterns or groupings in data without labels. The primary challenge with these algorithms is choosing the proper representation for the problem and input data. Because this is also a challenge for other types of machine learning it is discussed in a preceding section. Popular examples of unsupervised learning are clustering and autoencoders, often used in conjunction.

Clustering is useful where one wants to understand the fundamental types or classes within a group, so they can each be further characterized or understood. For example, in patient care, practitioners seeking patterns in Adverse Drug Reactions (ADRs) might cluster patients and look for those who do and do not experience ADRs within a class of drugs, such as antibiotics.[16] The emerging idiom of precision medicine – giving the right drug to the right patient at the right time – relies on an understanding of the underlying patient subtypes. Conversely, clustering is useful in situations where a maximally diverse representative sample needs to be drawn from a much larger space of examples. For instance, when performing a screen in a drug discovery project, for budget, throughput, or time reasons, a smaller representative set of compounds must be pulled from an internal library or purchased from an external vendor. Clustering can also be used to find more active compounds by seeding the clustering with previously known active compounds. A wide variety of different clustering algorithms are available, all with different strengths and weaknesses.

Autoencoders are a deep learning technique used to take high dimensional representations and distill them down to a more compact, lower dimensional representation. Autoencoders use two models during training, one to encode and one to decode. After training, the model used to encode the information is used to evaluate the test sets. Autoencoders may be used, for example, to analyze tumor gene expression data to look for tumor subtypes.[17]

---

16 Pinar, Y. et al., Knowledge discovery of drug data on the example of adverse reaction prediction, *BMC Bioinformatics*, **15** (Suppl 6):S7 (2014)

17 Rashid, S., et al., Dhaka: variational autoencoder for unmasking tumor heterogeneity from single cell genomic data, *BMC Bioinformatics*, btz095 (2019)

Semi-Supervised Learning

Semi-supervised learning shares the same objective as supervised learning, to build a function that will predict a label for a test set. The difference is that the training set doesn't have a complete set of features, thus they must be inferred. An example of a semi-supervised learning problem would be to build a model that predicts the best treatment option for a patient based on outcomes for different patients. Potential issues might be either the previous patients did not have all of the latest tests, or there are gaps in record keeping. The missing results need to be inferred from those patients in the training set for whom the test results exist. The missing data can be inferred before building a supervised machine learning model or it can be inferred as part of the model building process.

Generative Learning

Generative learning aims to create new examples drawn from the same distribution as the training set, and in some cases with a particular label. An example of generative learning is when novel molecular compounds need to be created with a set of desired properties. A generative model is first trained on a large set of molecules with known properties. The resulting model can then produce new molecules based on the input of the desired properties. This type of model can be useful in *de-novo* compound design in drug discovery.

Reinforcement Learning

Reinforcement learning is concerned with determining a set of actions that an actor must take in an environment to maximize a reward function. Most modern reinforcement learning techniques involve deep neural networks with some form of memory so that decisions are not made statically without context, but are based on the current state and information of previous states and decisions. An example of reinforcement learning is the development of models for optimal control based on a history of continuous glucose monitoring and insulin delivery for patients with Type 1 Diabetes.[18]

Evolutionary

Evolutionary algorithms are a set of machine learning methods inspired by evolutionary biology. Evolutionary algorithms start with a set of randomly generated examples. A loss function, often referred to as a fitness function, is evaluated on all of the examples. The examples that have the highest level of fitness are selected for 'reproduction.' A breeding function is used to perform crossovers and mutations to generate a new population of examples. The fitness function is evaluated on these examples and the process repeats until no more progress is being made in improving the fitness of the group. Evolutionary algorithms can be both generative when the example is the primary object, or discriminative when the example is a function used to estimate a training set of examples. Evolutionary algorithms have become more popular recently as a way to optimize the hyperparameters for other machine learning algorithms.

---

18 Yu, C. et al., Reinforcement Learning in Healthcare: A Survey. (2019) https://arxiv.org/abs/1908.08796v1

Active Learning

Active learning is a variant of supervised ML. Instead of having the model predict a measurable quantity it predicts which of the examples in a test set would be most informative to a subsequent supervised ML model. Active learning is often used to bootstrap a small training set into a larger training set for a supervised machine learning technique to learn from.

Transfer Learning

Transfer learning is a variant of supervised ML where information from a different, yet related, set of labeled training data are used to improve a model for a more specific set of labeled training data. An example of transfer learning would be the training of a model on a set of general labeled images, then using that model as the starting point for training a model to specifically perform facial recognition. When the original model is trained on a set of generically labeled images, the model learns the features that distinguish images (shade, texture, color, etc.). In the second round, model building requires less time on the basics of image recognition, but instead focuses on the specifics of facial structure that also include shade, texture, color, etc. In this way, the resources used to train a model (computational and training data) are conserved while making a model with better performance than one that was built de novo.

Multi-task Learning

Multi-task learning is often thought of as a type of transfer learning but is distinct. Here, multiple labels for the training set are used in building a supervised ML model, where the resulting model predicts all of the labels simultaneously. Not all training examples need to have all labels for training. This approach is only applied to datasets with labels that are in some way closely related. The advantage behind this approach is that if the labels are related then each label benefits from all of the training data, not just the subset of the training data that has that label. An example of multi-task learning is when trying to build a model of compound activity against a protein target with a small amount of training data. Instead of building a single-task supervised model to predict activity against the target, a multi-task model could be trained on all related proteins known to have homologous binding regions, thus improving the model predictions for the target with a limited amount of data.

Combinations/Hybrids

In many cases, multiple ML techniques are used together in an Intelligent Agent. One type of combination model is a consensus model, consisting of multiple sub-models. The sub-models generally use differing training set representations and different machine learning algorithms, such that they generate models with orthogonal strengths and weaknesses. Consensus models are often used in classification problems to reduce the number of false positives, but this is often at the expense of an increase in false negatives. There is almost always a trade-off. Another situation where one might combine techniques is in applying unsupervised learning techniques to determine which parameters should constitute the representation in a supervised model.

More complex Intelligent Systems can also be constructed using a variety of ML algorithms. For example, to predict off-target toxicity during drug discovery, models for off-target activity can be built and compounds of interest can be evaluated. For example, a knowledge graph may be built using natural language processing techniques applied to biomedical publications. The graph can be used to determine if the off-targets identified in the previous step are known to be associated with toxicity phenotypes. Combining ML models in this way can, however, lead to a multiplication of errors. Therefore, the various models must be analyzed to ensure that their errors are as independent as possible and must include an assessment of error propagation.

Hyperparameters

All machine learning algorithms have hyperparameters: variables that are not input data but govern how the algorithm builds the model. Examples of hyperparameters include the number of steps and the learning rate (for algorithms using a gradient search method to minimize the loss function over the training set), the number and the maximum depth of the trees (in a random forest model), or the size and number of layers (within a deep neural network). Hyperparameters are set either via previous human experience or by search. Searching for ideal parameters can be guided or done by brute force. Brute force search, often called grid search, generates a set of models based on a diverse set of hyperparameters spanning the range of parameter values. The hyperparameters associated with the model having the lowest final loss on the validation set are selected. More sophisticated search methods can involve other ML and statistical algorithms, such as Bayesian search or genetic algorithms, to find the optimal hyperparameters. This is often referred to as meta-learning.

Representation (Featurization)

With every ML problem, one must determine how to represent the data in the training set, e.g. the representation, also known as the feature set. The representation is the description of the objects in the training set, such as a photo or a set of measurements. Some algorithms require more sophisticated representations than others, but algorithms that require less sophisticated representations generally require larger training sets to train an equivalent model. Choosing the representation is vital to the ML algorithm's ability to detect a meaningful pattern in the training set.

The naive thing to do would be to include *everything and the kitchen sink* in the featurization of the training set. A training set is always finite, and thus the probability is non-zero that one of the features in the training set is correlated with the phenomenon being modeled. Because ML algorithms are not able to distinguish between correlation and causation, they may use this feature in the model. This leads to overfitting and reduced generalizability, as described in the appendix. When the model is then applied to a test set that doesn't contain this correlation the model will fail silently to be predictive.

A related, but often more difficult, aspect of feature selection is causality versus co-occurrence versus correlation. Again, because ML algorithms cannot distinguish between these

concepts, some understanding of the relationship between the representation and the training set labels must be present *a* priori. Or, controlled experiments could be designed to determine the relationship between the features in the representation and the labels.

There are two reasons why causal features are desired. First, if the features are only correlated there is a loss of generalizability, which will lead to a biased model, as all potential test sets will not contain the same correlations. Second, the long-term aim of applying ML to healthcare is to gain a better understanding of the causal structure underlying health and biology. While correlation, in some cases, and co-occurrence, in all cases, of features with labels will lead to predictive models, they will not lead to a better understanding of the phenomena underpinning that which is being modeled.

Unfortunately, it is often not obvious which features are best to use, because the phenomenon being modeled is not well understood, as is more often the case in biology. In these cases, there are methods to help determine the right representation, but often the only resort is trial and error using proper testing methods, preferably prospective testing.

Interpretability and Explainability

Explainable AI (XAI), also known as Interpretable AI or Transparent AI, refers to frameworks used to understand and explain the decision system within Intelligent Agents. This term has emerged due to the increasing complexity of Intelligent Agents generally. ML specifically yields difficult-to-interpret models—even for the data scientists who create them. ML algorithms may and often do arrive at predictions in a different way than humans. So, when a complex model makes a prediction, it may not be clear to humans *why* that prediction was made. Sometimes this obscurity is referred to as a "Black Box": data goes into the model and an output is produced – how it is produced remains a mystery to humans. For many applications, this is acceptable. If a retail store is trying to predict the total number of purchases for various products, it might not matter to any stakeholder how that prediction is made, provided it is accurate. In the healthcare industry, however, interpretability is of the utmost importance in many key areas. Being able to explain a model's predictions is essential for building trust and confidence in machine learning. With decisions about health, patients and doctors alike are appropriately reluctant to trust a decision-maker they cannot understand and cannot evaluate.

The challenge in developing Explainable ML is that it often requires a trade-off: foregoing a complex model with high accuracy that is difficult to explain, in favor of a less complex model that is easier to explain but with lower accuracy. Each situation will determine how to weight this seesaw, as described above. Explainability does not always imply lower accuracy, thus data scientists have to strike the right balance between explainability and performance in AI.

Machine learning models should be explainable in the following situations:

- When fairness is critical. For example, when patients are invited for cancer screenings (or not) based on a set of risk factors determined by the machine learning model, the selection criteria should be transparent.

- When consequences are far-reaching. Predictions returned by machine learning models can have far-reaching consequences in the healthcare industry (e.g., recommending that a patient have a risky operation, or classifying a malignant tumor).
- When transparency is required by law (e.g., the EU General Data Protection Regulation [GDPR], the Right to Explanation).[19]

Some of the models used to improve explainability include:

- REversed Time AttentIoN (RETAIN) Model[20]: Attempt to emulate a physician by training a modeling using an attention mechanism designed to give more weight to more recent EHR entries, as a physician might. Using RETAIN identifies the features and the visits that contribute most to the prediction.
- Local Interpretable Model-Agnostic Explanations (LIME)[21]: Take an existing, potentially very complex classification model and a prediction. From these it produces a simpler local model that can be interrogated.
- Layerwise Relevance Propagation (LRP)[22]: Work backwards through the model to find the relevance of each input.
- Distillation[23]: Train a simpler model such as a decision tree with a more complex model.

XAI is only now starting to produce results, therefore at least for a foreseeable future, XAI should be addressed by narrowing the tasks, using human control, and giving appropriate disclosure of the modeling approach. Due to high risk/benefit ratios in health-related decisions, reducing the scope is a better way of controlling model specificity than trying to dismantle the Black Box.

Fairness

A related topic to explainability is fairness. Fairness in ML receives a lot of attention nowadays as machine learning algorithms are being deployed to make critical decisions that impact people's lives, such as healthcare decisions. As with many ML concepts, fairness has a large number of definitions, many of which conflict with one another and many of which are context dependent. All measures of fairness focus on bias attributed to protected attributes, such as sex, gender or race.

Case law is one way to define fairness in big data and ML in the abstract.[24] More functional

---

19 Ahmad, M. A., et al., Explainable AI in Healthcare. (2018)
https://datamathstat.files.wordpress.com/2018/08/explainableaiinhealthcarekdd2018.pdf

20 Choi, E., et al., RETAIN: An Interpretable Predictive Model for Healthcare using Reverse Time Attention Mechanism, *30th Conference on Neural Information Processing Systems* (2016)

21 Ribeiro, M. T., et al., "Why Should I Trust You?" Explaining the Predictions of Any Classifier, *Knowledge Discovery and Data Mining Conference (2016)*

22 Bach, S. et al., On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation, PLoS ONE 10 (7)

23 Hinton, G., et al., Distilling the Knowledge in a Neural Network, preprint: arXiv:1503.02531v1 (Mar 2015)

24 Barocas, Solon & Selbst, Andrew D., Big Data's Disparate Impact, 104 Calif. L. Rev. 671 (2016)

definitions that can be used to test and correct for unfairness in ML have been explored by Gajane and Pechenizkiy (2018)[25] and Verma and Rubin (2018).[26] Some common concepts of fairness in the literature include:

- Fairness through Unawareness: Remove protected attributes from the training set representation and the model will achieve fairness.
- Group fairness: Both protected and unprotected groups having the same probability of being positive.
- Predictive Parity: Both protected and unprotected groups having the same probability of actually being positive when predicted to be positive.
- Equal Odds: Both protected and unprotected groups having the same probability of actually being positive when predicted to be positive and actually being negative when predicted negative.
- Counterfactual Fairness: Based on the causal structure of features, determines that if any feature used in the modeling is a downstream dependent feature on a protected attribute the model is not fair.

Fairness, no matter how it is defined, is of paramount importance in healthcare, both for protecting classes of individuals and providing safe and effective care. A full treatment of fairness in healthcare ML will be addressed in a future paper.

Machine Learning Techniques

Linear Regression

While often not thought of as a ML algorithm, linear regression does satisfy the requirements of a supervised ML technique when used for prediction. As such, it is not a very powerful technique because the set of model functions is restricted to only those of the form,

$$y = \vec{a} \cdot \vec{x} + b \tag{1}$$

Linear regression is generally used to produce a regression model and does so by minimizing a cost function that reduces the squared error between the prediction and the label (Mean Squared Error) over the training set by selecting a linear function ($\vec{a}$ and b in Equation 1) of the input features ($\vec{x}$).

Logistic Regression

Like linear regression, logistic regression is a very simple supervised ML method. Unlike linear regression, logistic regression is used to build a model for discrete labels, not contin-

25 Gajane, Pratik & Pechenizkiy, Mykola, On Formalizing Fairness in Prediction with Machine Learning, preprint: arXiv: 1710.03184v3 (May 2018)

26 Verma, Sahil & Rubin, Julia, Fairness Definitions Explained, ACM/IEEE International Workshop on Software Fairness (2018)

uous. Logistic regression uses the Cross-Entropy loss function to select a sigmoid function (model) of the input features (Figure 1). Despite "regression" in the method's name, the model generated by logistic regression is a classification model.

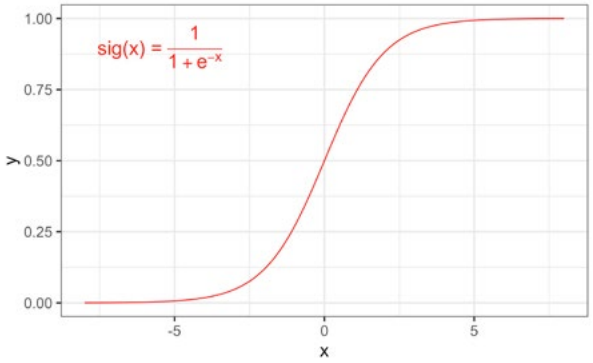

Figure *1*: Sigmoidal function used in logistic regression.

Decision Trees

A decision tree is a simple model that can be generated using ML techniques or basic heuristics. These can be developed to build classification, ranking, and regression models. A decision tree is a set of if-else statements that can be visualized as a tree (Figure 2). The leaf nodes for a classification decision tree are the classes. For other types of decision trees, the leaves are floating point numbers. Decision trees are constructed using a set of splitting criteria for determining which feature, or set of features, from the representation to split on at each vertex, and what value of the feature(s) will determine the split. The splitting criteria, the maximum depth, the pruning algorithm, and the decision about vertex order make up the bulk of the decision tree model building algorithm. One negative aspect of single decision trees is that they have a tendency to overfit. For example, the loss on this training set in Figure 2 is zero in that the model categorizes all points correctly. This may represent some overfitting and a pruning function that eliminates the level 4 decision of x < -1.5 will result in a higher training loss, but a less complex model, and a better prediction error. This is a situation where a holdout set would help in determining and correcting the overfitting to some degree.

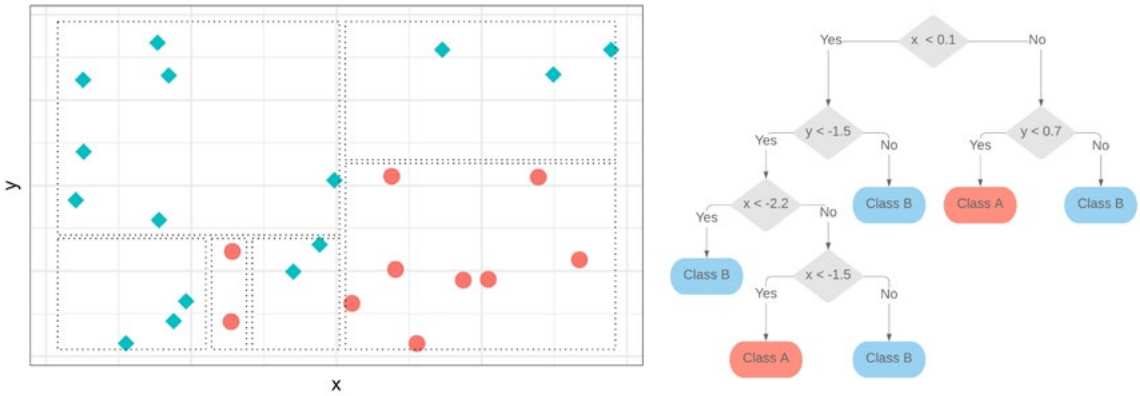

Figure *2*a: Partition boundaries for a decision tree using univariate vertex decisions. *2*b: Decision tree details.

Random Forest (RF)

As the "forest" it its name implies, a Random Forest model, introduced in 1995,[27] is a single model made up of a set of decision trees. The Random part of the name comes a component of stochasticity in all RF algorithms to help reduce the overfitting problem found in decision trees. Where the randomness is introduced depends on the Random Forest algorithm used. The modern tree building algorithms within a RF model builder use random subsets of the training set with replacements for building each tree, using the holdout set as a validation set for pruning, and it restricts the algorithm to a random subset of the input features to select from at each decision point or vertex. The function used to determine the feature and split at each vertex varies. Random Forest algorithms generally use majority rule when combining outputs from classification trees and some form of mean calculation when combining outputs from regression trees. Random Forest models are one of the most widely used model types today, generally and in the field of healthcare.

Support Vector Machines (SVM)

Support Vector Machines treat the input data features as a vector in a high dimensional feature space. For binary classification the algorithm attempts to find the hyperplane that minimizes the loss by maximizing the margin. This is the minimal distance between the hyperplane and the closest data points in the two classes (Figure 3). When data are not well separated by a flat hyperplane, a mathematical function called a kernel can be used to transform the data such that a flat hyperplane separates the data well. This has the effect of fitting the data with a more complex model function. As was previously discussed in the Model Performance section, the more complex the kernel function, the more complex the model and the higher likelihood of over fitting. In Figure 3 it is clear that there is some training error in the model in that there are some red points above the model line. By transforming the data with a kernel or equivalently using a more complex non-flat hyperplane for a model, the training loss can be reduced to zero.

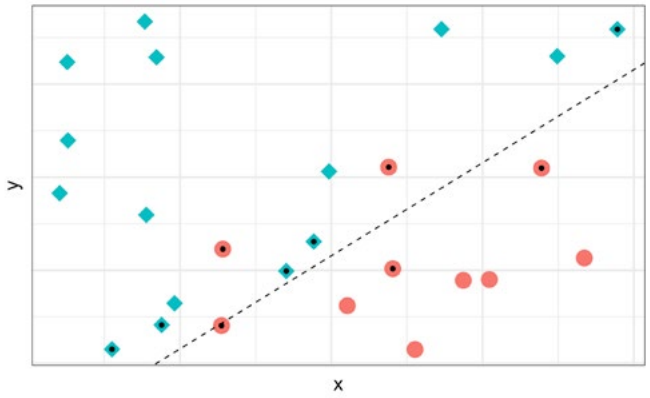

Figure *3*: An SVM model, represented by the dashed line, was built using the same training set as in Figure *2*a. The support vectors in each class are identified by inset black dots.

27 Ho, T. K., Random Decision Forests. *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, pp 14-16 (Aug 1995)

Artificial Neural Networks (ANN)

The computational model for neural networks is based on work presented by Warren McCulloch and Walter Pitts in 1943.[28] Artificial Neural Networks are networks of artificial neurons, inspired by the connectivity architecture of neuron cells in the brain. A neuron, in this case, is a mathematical function that takes a set of inputs from an input layer, or previous layer in the network. The inputs are then modified by weights associated with each input and summed. The sum is then input into an activation function, such as in Figure 1, with an output normally between -1 and 1 or 0 and 1. This output is then fed into the next layer of neurons or the output layer (Figure 4). The ANN model variables are the weights associated with each input to each neuron. For supervised learning models using ANNs the weights are generally adjusted using what is known as backpropagation, developed in the 1970s.[29] There are many types of ANNs, such as modern deep neural networks and convolutional neural networks for supervised machine learning, and autoencoders and Boltzmann machines for unsupervised machine learning.

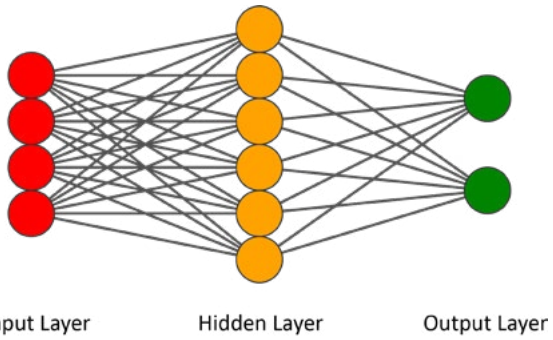

Figure *4*: Example of a fully connected ANN with one hidden layer, four input features, and two output results.

Deep Neural Networks (DNN)

A Deep Neural Network is an ANN with multiple layers between the input and the output. Until fifteen years ago the computational power wasn't available to perform training of DNNs over reasonably large datasets. This is one of the underlying reasons why DNNs have surged in popularity and usefulness recently while much of the framework was develop more than 40 years ago.

Convolutional Neural Networks (CNN)

Convolutional Neural Networks are currently one of the most important deep learning methods, and are attributed with much of the recent resurgence of neural network usage for data analytics. AlexNet, an image classification model, was one of the early examples of the power of CNNs.[30]

---

28 McCulloch, W. and Pitts, W., A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Math. Biophysics*, **5** (4), (Dec 1943)

29 Werbos, P. J., *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Harvard University Press (1975)

30 Krizhevsky, A. et al., ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems 25* (2012)

The input of a CNN consists of data that can be efficiently represented in a Euclidean fashion (e.g. arrays or grids). A good example of this is an image that can be represented as a 3D array, where the third dimension represents the color channels. The CNN filtering layers consist of a set of convolution, pooling, action potential and sub-sampling layers (details of which go beyond the scope of this document) where spatially close sections of the inputs are combined together in a repeating pattern which generally takes a wide-thin 3D array and transforms it into a narrow-long array (Figure 5). This narrow-long array of numbers is then passed into fully connected DNN. The advantage of using these filtering operations instead of fully connected methods in the first few layers is that it reduces the dimensionality of the input while generating higher order features. The features that emerge from the filtering layers during the training phase often eliminate the need for a priori designed features.

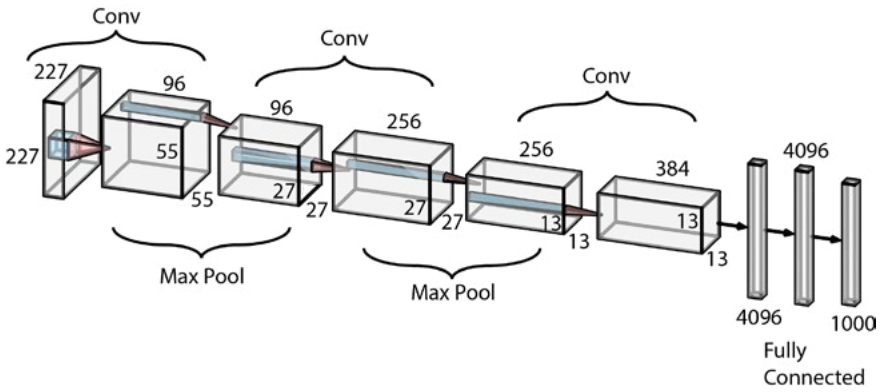

Figure 5: An example CNN with 3 convolution layers, 2 max pool layers (alternating) and fully connected convolution network. The input is an image of size 227x227 with 3 color channels and the output is a probability vector of size 1000.

The patterns that result due to the filtering are inspired by biological processes and are thought to be similar to how neurons in the visual cortex are organized. They only respond to inputs from a restricted region, known as the receptive field. Because the input is based on grids, it allows the CNN to capture both spatial and temporal dependencies, which is what makes a CNN such an effective method for image processing. The filters will, by themselves, learn to detect certain data features such as textures, edges, corners, points of interest, etc. Although CNNs were originally developed for image-based operations such as recognition and classification, recently the methods have also successfully been applied to natural language processing and speech recognition.

Graph Convolutional Neural Networks (GCNN)

Whereas the previous section described how CNNs operate efficiently on Euclidean data (e.g. grids), such as images, they are unsuitable for non-Euclidean data, such as graphs. The complexity of non-Euclidean data poses several challenges for standard ML methods and requires its own class of solutions. One such solution is a specialized convolution operation that works on graph-based input data.

In graph-based networks the nodes of the graph are related to their neighbors via some complex linkage information that captures the interdependence among data, something

which is not possible using standard ML methods. GCNNs efficiently model this interrelation between the graph nodes and are able to process the relations between the nodes using specialized convolution operations. GCNNs are useful for node embeddings, graph classification, knowledge graphs and graph generation.

GCNN approaches fall into two categories:

- **Spectral GCNN:** This approach defines the graph convolution by introducing filters from the perspective of graph signal processing where the graph convolution operation is interpreted as removing noise from the graph signals.
- **Spatial GCNN:** This approach formulates the graph convolutions as a feature that aggregates information from connected neighbors (Figure 6).

In healthcare, GCNNs are used for structure activity relationship (SAR) models of chemical compounds, where the graph inputs are molecular graphs and the properties are atom based.

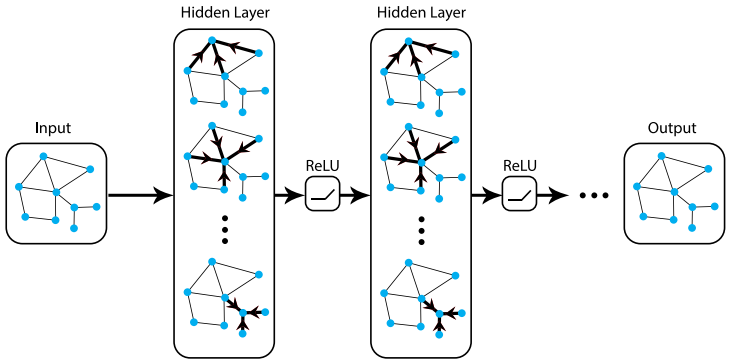

Figure *6*: Spatial graph convolutional neural network representation.

Generative Adversarial Networks (GANs)

Generative Adversarial Networks were originally proposed by Ian Goodfellow et al. in 2014.[31]

Bridging deep learning and game theory, GANs are used to generate or "imagine" new objects with desired properties. Since 2016, multiple implementations of GANs architecture in combination with reinforcement learning have been applied to de novo molecular design,[32] medical image processing,[33] and other important tasks in the health sciences.[34] A brief timeline of GANs development is presented in Figure 7.

31 Goodfellow, I. J. et al., Generative Adversarial Networks, preprint: arXiv:1406.2661v1 (Jun 2014)

32 Kadurin, A. et al., druGAN: An Advanced Generative Adversarial Autoencoder Model for de Novo Generation of New Molecules with Desired Molecular Properties in Silico. *Mol Pharm*, **14** (9), pp. 3098-3104 (2017)

33 Kazeminia, S. et al., GANs for Medical Image Analysis. Preprint: arXiv:1809.06222v2 (Sep 2018)

34 Anand, N. & Huang, P., Generative Modeling for Protein Structures. *32nd Conference on Neural Information Processing Systems* (2018)

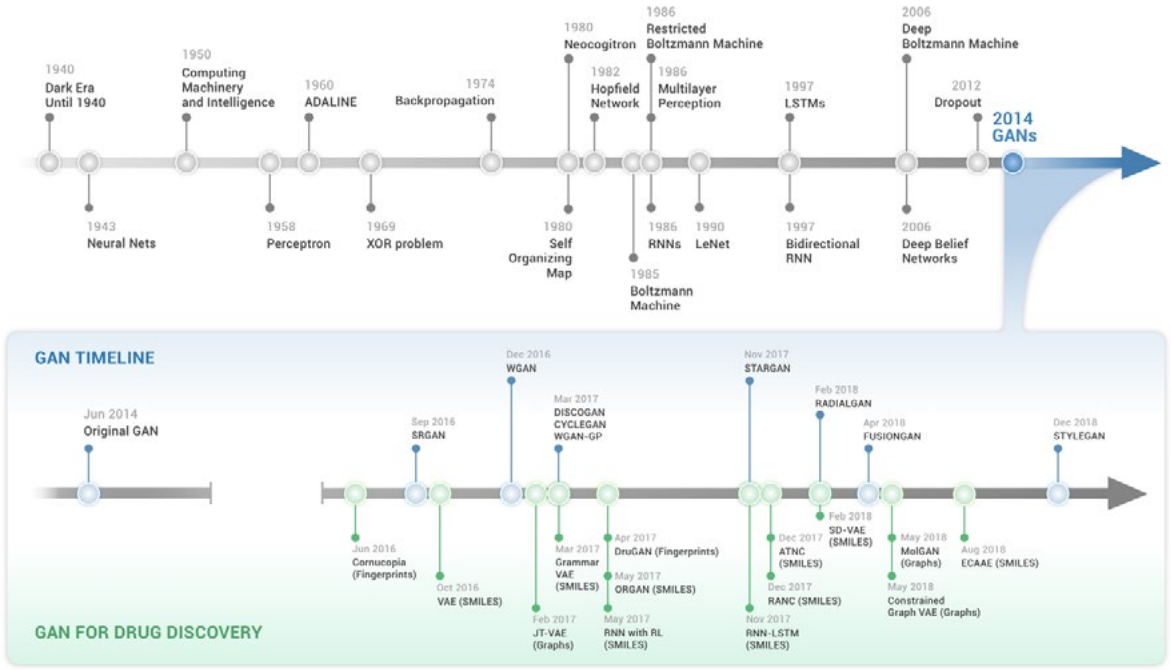

Figure *7*: GANs and the history of deep learning.

GAN architecture consists of two deep neural networks—a generator and a discriminator—that are trained together. During the training phase of the GANs, the generator and the discriminator play a game where the generator learns to produce examples that imitate the training set and the discriminator tries to identify which examples are generated and which are from the training set. The simplest GAN architecture is presented in Figure 8. One of the primary difficulties with GANs is getting them to converge to a stable set of networks and not having them collapse (the network always produces the exact same output, independent of the input) or become unstable (the network produces different outputs for the exact same input). More complex architectures briefly mentioned in Figure 7 are used to avoid these issues as well as achieve various goals. In the simple architectural example in Figure 8, once the models are trained in the case of *de-novo* molecule design or protein fold prediction, the generator can be used to generate new hypothetical compounds or foldings for a protein, respectively. For image processing combinations, the generator and discriminator can be used for various tasks, including synthesis, segmentation, reconstruction, detection, de-noising, registration, and classification.

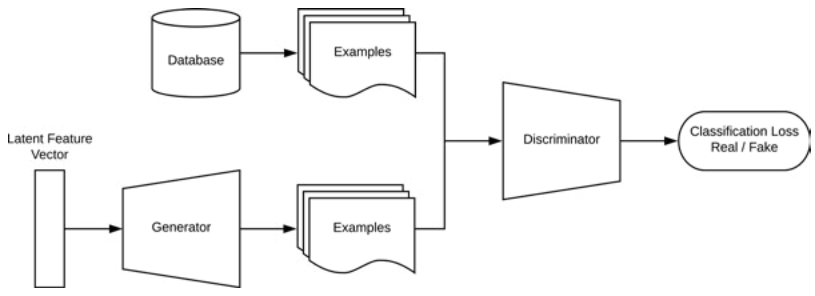

Figure *8*: GAN-RL training diagram using reinforcement learning. The error is propagated back from the loss through the discriminator and generator.

While GAN and GAN-RL techniques are currently state-of-art in deep learning, they have many limitations and require expert knowledge of topics they are being applied to, such as chemistry and biology for molecular generation. Firstly, the training sets used for generative systems need to be balanced and properly annotated. Training sets like this are hard to find in any field. Secondly, achieving diversity while generating valid examples is very difficult as there is generally a tradeoff between the many properties needed in an example.

RELATED APPROACHES

There are many other ways to generate an Intelligent Agent that uses a deductive approach, either on its own or in combination with inductive Machine Learning. This section examines a number of noteworthy approaches that may or may not be familiar to the reader.

Heuristic

According to the International Encyclopedia of the Social & Behavioral Sciences, Heuristics are "approximate strategies or 'rules of thumb' for decision making and problem solving that do not guarantee a correct solution but that typically yield a reasonable solution or bring one closer to hand. As such, they stand in contrast to algorithms that will produce a correct solution given complete and correct inputs. More specifically, heuristics are usually thought of as shortcuts that allow decisions or solutions to be reached more rapidly and in conditions of incomplete or uncertain information—often because they do not process all the available information."[35]

Heuristics can lead to cognitive biases, and there is a discord between bias and rationality.[36] In the context of AI, if an Expert System or an AI is built by humans (or another AI) using heuristics, then they may be encoded with bias and error, as well. Heuristics do not always lead to the most optimal (or fair) outcome. That being said, heuristic algorithms can be much faster than traditional algorithms, and often use considerably less computational power.[37] Heuristics might be built into an AI such that it limits the solution space to be searched.

Building heuristic models requires very demanding processes, based on set of rules (derived from initial data analysis) that are simpler to explain and understand than ML models. However, since Heuristic methods in most cases use experimentation and trial-and-error techniques that require "rigorous definition, careful collection of data, and thorough and disciplined analysis, it places immense responsibility on the researcher."[38]

---

35 Todd, P. M., Heuristics for Decision and Choice, *International Encyclopedia of the Social & Behavioral Sciences*, Elsevier Ltd. (2001)

36 Heuristic, Behavioral Science Solutions Ltd, 2014-2019, https://www.behavioraleconomics.com/resources/mini-encyclopedia-of-be/heuristic/

37 101 Computing, Heuristic Approaches to Problem Solving. (Feb 2018) https://www.101computing.net/heuristic-approaches-to-problem-solving/

38 Frick, Willard B, The symbolic growth experience: A chronicle of heuristic inquiry and a quest for synthesis. Journal of Humanistic Psychology, 30(1), 64-80

Optimization

Mathematical optimization is the search for input parameters to an objective function that yields maximum or minimum output values. Optimizations are widely used in ML. However, the optimization itself is not fundamentally a subdiscipline of ML. Optimizations are broadly used in non-ML applications, including many sectors of scientific computing. Optimizations can also be used to combine multiple predictive tasks into single, higher-order decisions which require balancing tradeoffs between multiple favorable properties. This task is known as *multi-objective optimization*. When applied to decision making, algorithms that address multiple objectives could be considered in the same category as *heuristic solutions*, providing a non-ML based expert system.

Scientific Modeling and Simulations

Scientific Modeling is the practice of representing real-world systems as mathematical abstractions that can be used to generate new understanding of complex systems. Modeling is often coupled with simulations to experiment with varied scenarios and make predictions on future outcomes. Simulations transition a model from a starting state through a successive series of states. While these computational methodologies yield predictions, they are not considered ML subdisciplines since they are deductive in nature, not inductive.

Nonetheless, Simulations and ML have many shared attributes and are often combined in an Intelligent Agent. For instance, predictions generated by simulation may be evaluated using the same metrics as supervised learning tasks. Models and simulations may also have parameters that are experimentally set by reconciling predictions to real world observations. However, the underlying functions driving a simulation from one state to the next are derived by experts based on a scientific understanding of the real-world processes or established statistical distributions.

The concepts of Simulations and ML have become increasingly intertwined as practical applications become more and more complex. For instance, reinforcement learning is a form of simulation whereby the function governing state-to-state transitions is derived through ML. In reinforcement learning, training data is generated in real time on a trial-and-error basis using an external objective function to define success. Alternatively, supervised learning models can be used to define objective functions or target states for simulation-based tasks. For example, DeepMind's alpha-fold pipeline uses a combination of simulation-based techniques and ML to predict the 3D atomistic structure of proteins directly from their amino acid sequences. This hybrid approach uses simulated annealing (see below) to generate candidate structures and two discriminative neural networks to evaluate them on the basis of inter-residue distances and general similarity to select properties of experimentally derived protein conformations.[39]

39 Evans, R. et al., De novo structure prediction with deep-learning based scoring, Thirteenth Critical Assessment of Techniques for Protein Structure Prediction (Abstracts) (Dec 2018)

Monte Carlo Simulations and Simulated Annealing

In Monte Carlo Simulations, state-to-state transitions are determined stochastically based on observed statistical distributions,[40] rather than determined through a scientific understanding of underlying processes. These simulations can be used to estimate complex probabilities or explore states of a complex feature landscape. Simulated Annealing is an optimization protocol based on Monte Carlo simulations, whereby each individual state-to-state transition is accepted or rejected on the basis of an acceptance function. Transitions are accepted when the new state is considered favorable in accordance with the reward function. Less favorable state transitions are randomly accepted in accordance with a probability that decreases over time. Occasionally accepting unfavorable state transitions helps the simulation escape local minima when searching a global landscape of solutions. Molecular Docking is a notable example of simulated annealing in computer-aided drug design, where molecular affinity potentials derived through statistical approaches gradually guide the 3D coordinates of small molecule ligand into a protein binding site.[41] Molecular docking is used to predict the 3D coordinates of protein-ligand complexes, and ranking libraries of molecules by their likelihood of binding a target protein.

FURTHER TERMS

Ontology

In the context of information sciences and, by extension, AI, an ontology is "a formal description of knowledge as a set of concepts within a domain and the relationships that hold between them. To enable such a description, we need to formally specify components such as individuals (instances of objects), classes, attributes and relations as well as restrictions, rules and axioms. As a result, ontologies not only introduce a shareable and reusable knowledge representation but can also add new knowledge about the domain."[42] In short, an ontology is a framework to represent information. A viable framework must provide AI with the knowledge or ability to understand, reason, plan, and learn with datasets, and must generate reproducible results.

Data quality is of the foremost importance for building accurate ML models. Any ambiguity may result in skewed predictions by the algorithm. By using ontology-based data cleaning as a pre-processing step, the model can better understand data input and build more accurate models. For example, mutations in the human gene BRCA1 can lead to breast cancer. This same gene has the official synonym 'IRIS'[43], which is also a name for part of the eye.

---

40 Hastings, W.K., Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, **57** (1) pp. 97–109 (1970)

41 Goodsell, D. S., & Olson, A. J., Automated docking of substrates to proteins by simulated annealing. *Proteins: Structure, Function, and Bioinformatics*, **8** (3), pp. 195–202 (1990)

42 Ontotext, What are Ontologies? (2019) https://www.ontotext.com/knowledgehub/fundamentals/what-are-ontologies/

43 Harland, L., Are Ontologies relevant in a Machine Learning-centric world? (OCT 2018) https://www.scibite.com/news/are-ontologies-relevant-in-a-machine-learning-centric-world/

When working with data containing genomic information (as an input or output variable), ontology-based data pre-processing should thus indicate that the term 'IRIS' be treated as a gene and not a term relating to the anatomy of the eye or a flower. In this way, current domain-specific ontologies might be used to bolster an AI's understanding, learning, and ability to make predictions within that domain. The Gene Ontology[44], which has been designed to capture all current knowledge about the function of genes, is an example of one such domain-specific ontology that could strengthen an AI in the human genomics space.

Robustness

Robustness is the concept that a machine learning algorithm and produced models are stable with respect to small changes in the training set, such as the addition of new data or simply a reordering of the training set examples. Many machine learning algorithms unfortunately are not very robust, therefore a technique of ensembling is often employed.

Ensembling

Ensembling is a technique that combines a number of models with the introduction of random variations, either in the training set, initial conditions, etc. The outputs of each of these models are then combined via averaging or some other method producing a more stable output, and may obtain better performance than could be achieved from any of the member algorithms alone. That is, the group of algorithms may each be well-suited to some parts but not all of the problem space. Some ML algorithms, such as random forest, have Ensembling built into them.

Meta-Overfitting

Meta-Overfitting can often complicate the use of a "golden test set" to benchmark model performance. A golden test set is generally used as a measuring stick to compare ML models and is meant to be a good representative of the background space or the entire test space. As such, the information in the golden test set is meant to be independent of the model. However, if the hyperparameters and feature sets are optimized solely on the grounds of performance on the golden test set, the information in the golden test set is no longer independent, because it has been used to inform the model. Two ways to avoid meta-overfitting is by randomly selecting a subset from the golden dataset to be used in each evaluation, and by establishing whether an improvement against the golden test set is statistically significant.

Small Data and AI

A successful AI healthcare system can be built with millions upon millions of data points, down to just a few hundred data points ("small data"). However, the transition from big data to small data is one of the key trends shaping the way healthcare companies are building AI/

44 Gene Ontology Consortium, Gene Ontology Resource, http://geneontology.org/

ML models. This trend toward "small data" is a result of necessity rather than design, due to limited access to healthcare data, e.g., few recorded events, rare diseases, lack of consistent channels of information, etc. Elsevier Ltd. published an article in 2018 titled *"Using deep neural network with small datasets to predict material defects*" that demonstrates that DNN (deep neural network) with small datasets and pre-training can be a reasonable choice when big datasets are unattainable for specific use cases in healthcare.[45]

A "meaningful small data" approach in healthcare means driving towards lean AI/ML models, incremental data infrastructure investments, and emerging ML approaches. These strategies can lead to better results in use cases such as treatment variability, clinical trial eligibility, drug utilizations, etc.[46]

Trustworthiness

To fully realize the benefits of AI in healthcare, we need transparent and trustworthy AI solutions which are interpretable, and that support specific business and patient care needs. The following guidelines will help improve the trustworthiness of an AI system in healthcare:

- Use exploratory data analysis before model building.
- Identify outliers in the data and generate a set of likely outcomes from the training dataset to verify it with model outputs.
- Store data from each possible intermediate stage or layer of the machine learning process, and utilize the model agnostic framework to explain local results in order to identify decision-making mechanisms by learning algorithms.
- Validate the quality of public datasets. Datasets should be shared, but there should also be a process to validate and vet the quality of the data.
- Evaluate key factors that may affect our judgment of trustworthiness. This includes: support industry standards (e.g., ISO, IEEE), accuracy, security, data privacy, ethical standards associated with AI, standardization on the outcome of an AI solution, etc.
- Communicating AI standards to society will play a key role in public trust of AI technologies. To that end, the High-Level Expert Group on AI (AI HLEG) published a set of Ethics Guidelines to promote Trustworthy AI solutions.[47]
- Society tends to be discriminatory of AI systems. If we are not careful, a lack of trust could perpetuate or increase that discrimination regardless of whether a solution delivers the expected outcome or not.[48]

---

45 ScienceDirect, Handling limited datasets with neural networks in medical applications: A small-data approach, https://www.sciencedirect.com/science/article/pii/S0933365716301749

46 H1insights, Part 1: Predicting Healthcare Trends of 2019, https://learn.h1insights.com/blog/2018/11/15/part-1-healthcare-trends-for-2019

47 European Commission, Ethics Guidelines for Trustworthy AI, https://ec.europa.eu/futurium/en/ai-alliance-consultation

48 The Conversation, The Montréal Declaration: Why we must develop AI responsibly, https://theconversation.com/the-montreal-declaration-why-we-must-develop-ai-responsibly-108154

EXAMPLES OF INTELLIGENT AGENTS IN HEALTHCARE

A large and rapidly growing number of examples of Intelligent Agents are being used in healthcare. Below are selected use cases demonstrating how Intelligent Agents powered by ML techniques are already impacting Healthcare.

Real-Time Septic Shock Warning

Background

Sepsis, a clinical syndrome of life-threatening organ dysfunction caused by a dysregulated response to infection, is a leading cause of death in the United States, with mortality highest among patients who develop septic shock. Septic shock is an extreme case of sepsis where a patient experiences dangerously low blood pressure and abnormalities in cellular metabolism. Morbidity, mortality, and length of stay are greatly reduced by early detection and treatment of septic shock.

Case-Study

Approach

Henry et al. at Johns Hopkins University developed a real-time early warning score (TREWScore) for septic shock.[49] Using electronic health records (EMRs) from 13,014 sepsis patients (1,836 who developed septic shock and 11,178 who did not) a Cox proportional hazards model was used as a supervisory signal using time course data for 54 parameters, such as blood pressure and white blood cell count. The new model was trained with this signal to estimate the time to an adverse event.

Solution

A set of 3011 randomly selected patients (455 who developed septic shock and 2556 who did not) were held out as a validation set. For each patient in the validation set the time course data was played forward and the TREWScore was recomputed for each new time point. A patient was identified as at risk when his or her score crossed the specified risk threshold. In the validation set, the AUC obtained for the TREWScore was 0.83 (95% CI, 0.81 to 0.85) (Figure 9).

49 Henry, Katharine, A targeted real-time early warning score (TREWScore) for septic shock, Science Translational Medicine, Aug 2015

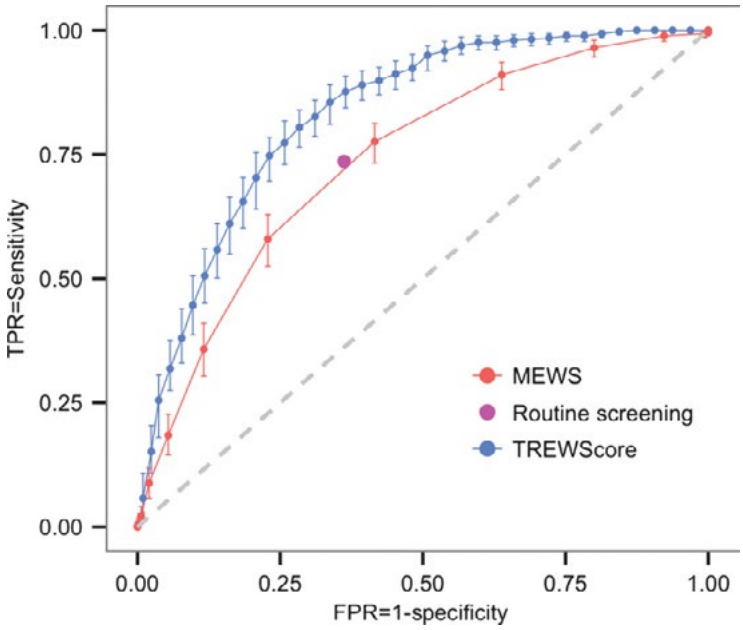

Figure *9*: Receiver Operator Curve (ROC) for detection of septic shock before onset in the validation.

At a specificity of 0.67 [false-positive rate (FPR) of 0.33], TREWScore achieved a sensitivity of 0.85. Patients were identified at median of 28.2 hours (IQR, 10.6 to 94.2) before shock onset. This result was compared to a Modified Early Warning Score (MEWS) for clinical deterioration, which is a simple, physiological score that allows improvement in the quality and safety of management provided to surgical ward patients.

Business Results

The expanded use of EMRs, as well as greater access to continuously captured vital measurements, has created a valuable opportunity for real-time ML solutions. Earlier identification and treatment of patients likely to experience septic shock can dramatically reduce morbidity and mortality, saving lives and reducing the amount of time in the hospital. With healthcare shifting its focus to outcomes-based reimbursement, preventing septic shock could have a major economic impact.

Skin Cancer Classification

Background

Skin cancer is the most common human malignancy and is diagnosed visually beginning with an initial clinical screening, typically followed by dermoscopic analysis, and if suspicious, a biopsy which results in histopathological examination. One in five Americans will be diagnosed with a cutaneous malignancy in their lifetime. Although melanomas represent fewer than 5% of all skin cancers in the United States, they account for approximately 75% of all skin-cancer-related deaths and are responsible for over 10,000 deaths annually in the US.

Dermatologists are not available in all regions of the USA and are costly so many patients may opt not to visit a dermatologist. Primary care providers may have limited skills in this area and may overlook cancers. As a result, being able to train a ML model to detect cutaneous lesions and classify as malignant or not would be the first step to diagnose skin cancer using a mobile device. This could lead to greater access to the visual screening component, potentially saving thousands of lives in the US alone.

Case-Study

Approach

Esteva et al. at Stanford implemented a skin cancer classification system using a deep convolutional neural network learning approach to solve the problem of automated cutaneous malignancy diagnosis based on images.[50] The input training set consisted of 129,450 clinical images labelled with 2,032 different diseases. The CNN model that was developed was compared to the opinion of experts on previously unseen images.

Solution

Data flows from left to right in Figure 10. An image of a skin lesion is sequentially warped into a probability distribution over clinical classes of skin disease using Google Inception v3 CNN architecture pretrained on the ImageNet dataset (1.28 million images over 1,000 generic object classes) and fine-tuned on the training set of 129,450 skin lesions comprising 2,032 different diseases.

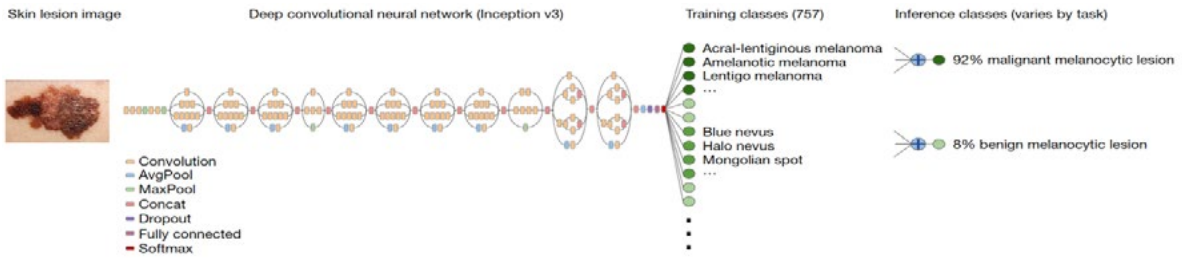

Figure *10*: Neural network architecture including example input image and resulting classification.

Business Results

The accuracy of the trained CNN in detecting malignancies matched that of 21 trained dermatologists. The CNN also matched the trained dermatologists in the ability to identify the lesion class. This enables fast detection and classification of skin cancer as it empowers primary care practitioners to perform initial screening. The model could even be used in non-medical settings when it is deployed on a mobile device.

50 Esteva, A. et al., Dermatologist-level classification of skin cancer with deep neural networks., *Nature*, **542** (7639), pp.115–118 (2017)

Pharmacovigilance

Background

Pharmacovigilance (PV) is the study of adverse effects caused by pharmaceutical products. Almost all markets for pharmaceutical products have regulations around collecting and studying PV data, both pre- and post-approval. With the increase in human longevity, increased access to pharmaceuticals, and emergence of internet-connected monitoring devices, the amount of PV data being processed by companies related to their products is increasing rapidly.

Case-Study

Approach

Pfizer concluded that case processing activities (*extracting data using Natural Language Processing from submitted documents*) constitute up to two-thirds of the internal PV resources. In order to improve the cost and efficiency of case-processing they compared PV AI solutions from three vendors and their internal AI Center of Excellence.[51]

Solution

In the pilot, the vendors' AI systems were trained using a set of 50,000 correctly annotated documents. The systems were then tested on 5,000 unannotated test documents and then compared with the hand processed annotated version. In the second cycle, the amount of training data was doubled to establish if the systems would become more accurate as more training data was used (Figure 11).

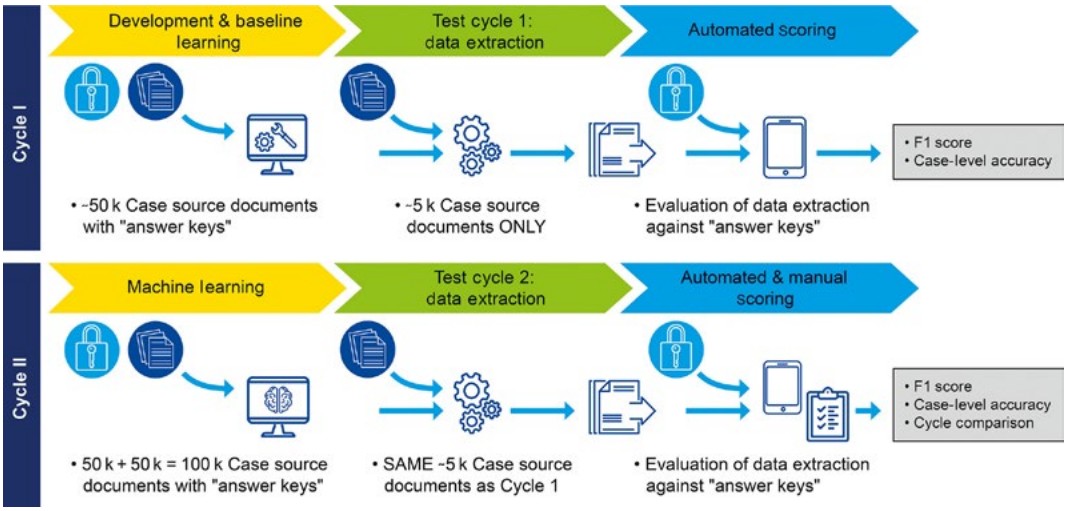

Figure *11*: Pilot system set-up.

51 Schmider, J. et al., Innovation in Pharmacovigilance: Use of Artificial Intelligence in Adverse Event Case Processing, *Clin Pharm Thera*, **4** (Apr 2019)

Business Results

The study found that two of the vendors had an accuracy rate of over 70% in extracting information from PV data test sets. The case-level accuracy showed that the same two vendors were able to process 30% of cases with a greater than 80% accuracy and showed improvement in the second cycle.

Small Molecule Drug Discovery

Background

Small molecule drug discovery is a design-make-test process to find an optimal candidate molecule out of the 10^{60} theoretical small molecules. The resulting compound must satisfy a very large set of criteria including: potency against the primary protein target, efficacy in animals, ADMET (absorption, distribution, metabolism, elimination and (lack of) toxicity). According to the Pharmaceutical Research and Manufacturers of America (PhRMA) this preclinical process can take 3-6 years and cost more than $25 million. Only one out of four programs makes it into clinical studies, and fewer than 12% of those candidates ultimate reach approval by the FDA.

Case-Study

Approach

The drug discovery process searches for highly active compounds that also have acceptable biochemical and toxicity properties (ADMET). Traditionally, medicinal chemists study existing compounds and their associated activity and ADMET assay data, known as the Structure-Activity Relationship (SAR). From these data, patterns arise which lead to the discovery of more active compounds. However, this process of finding more active compounds involves laborious synthesizing and testing large numbers of compounds as most of which are ultimately ineffective, as they do not lead to compounds with the required properties and criteria.

Solution

There are many machine learning solutions being applied to the problem of pattern recognition within SAR data. In fact, this is one of the first areas in drug discovery where machine learning has been applied. The earliest techniques for building machine learning models used engineered feature sets:

- Calculated molecule properties, such as lipophilicity, polar surface area, etc.;
- 2D molecular fingerprints based on the presence of particular sets of atoms or on particular atom-bond-atom pathways in the molecule;
- 3D molecular fingerprints based on the presence of particular features and the 3D dis-

tances between them for a given conformation of the molecule, also known as, pharmacophores;

- Some combination of the above features.[52]

The machine learning techniques used with these engineered feature sets also varied with the favored algorithms being Random Forest or Deep Neural Networks. With the introduction of graph convolutional neural networks, the ability to build a model without selecting a specific set of engineered features was introduced. The only caveat being that generally a larger training set is required compared to methods using engineered features. Some progress has been shown since using transfer learning.[53]

Business Results

A typical high throughput screen (HTS) results in 0.1% hit-rate from a screening library from 10,000 to one million compounds in size. With machine learning models, virtual spaces of over one *billion* synthetically accessible compounds (synthesizable in under four weeks for less than $100/molecule) can be performed. Hit rates for compounds selected from such libraries based on machine learning model predictions have been shown to be as high as 30-50% while the hits remain diverse and unique related to the training set.

In the lead optimization phase of drug discovery—where large spaces of de novo compounds are being searched using multiple machine learning models of activity and ADMET—the use of such models can greatly increase the efficiency of the design-make-test cycle leading to an order of magnitude (OOM) fewer compounds synthesized and tested in a fraction of the time. Theoretically, a lower failure rate in the clinic should also be observed since an OOM more compounds were examined against an OOM more virtual assays than could be performed traditionally in the pre-clinical phase.

De Novo Small Molecule Generation

Background

As stated above, small molecule drug discovery requires a vast search space and a large number of design-make-test cycles to find a single candidate molecule. Candidates must meet numerous stringent criteria for in vitro and in vivo characteristics and behavior. Producing, or "inventing," molecules with optimal drug like properties instead of searching for one is exactly the idea being pursued by a number of groups using GANs. The first peer-reviewed publications on the application of GANs to generative chemistry utilized molecular finger-

52 Yang, K., et al., Are Learned Molecular Representations Ready for Prime Time? preprint: 10.26434/chemrxiv.7940594.v2 (Jul 2019)

53 Altae-Tran, H., et al., Low Data Drug Discovery with One-Shot Learning. *ACS Cent Sci*, **3** (4), pp. 283–293 (2017)

print representation and an adversarial autoencoder (AAE) architecture[54,55] and string-based representations using the variational autoencoders (VAE).[56,57] Later versions combined GANs and reinforcement learning by introducing the Objective Reinforced Generative Adversarial Network (ORGAN) architecture for generation of novel molecules[58], the approach further extended by introducing the adversarial threshold (AT)[59] and differential neural computer (DNC)[60] concepts. GANs have been shown capable of generation of novel molecules using the graph representation of the molecular structure[61] and 3D representations.[62]

Case-Study

Approach

Zhavoronkov et al. developed a generative tensorial reinforcement learning (GENTRL) neural network for *de novo* lead-like molecules design with high potency against the protein of interest.[63] They were able to discover potent inhibitors of discoidin domain receptor 1 (DDR1), a kinase target implicated in fibrosis and other diseases, within 21 days.

Solution

Reinforcement learning, variational inference, and tensor decompositions approaches were combined into a generative two-step machine learning algorithm named GENTRL. The first step was to train an autoencoder-based model to learn a mapping from discrete molecular space to a continuous parametrized space. At this stage the network was trained on the general medicinal chemistry dataset together with kinase chemistry and patent data. MCE-18, IC50 and medicinal chemistry filters were used to parametrize learned representation.[64]

---

54 Kadurin, A. et al., The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology. *Oncotarget*, **8** (7), pp. 3098-3104 (2016)

55 Kadurin, A. et al., druGAN: An Advanced Generative Adversarial Autoencoder Model for de Novo Generation of New Molecules with Desired Molecular Properties in Silico. *Mol Pharm*, **14** (9), pp. 10883-10890 (2017)

56 Gomez-Bombarelli, R. et al., Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Cent Sci* **4** (2), pp. 268-276 (2018)

57 Lim, J. et al., Molecular generative model based on conditional variational autoencoder for de novo molecular design. *J Cheminform* **10** (1), pp. 31 (2018)

58 Benjamin, S.-L. et al., Optimizing distributions over molecular space. An Objective-Reinforced Generative Adversarial Network for Inverse-design Chemistry (ORGANIC). preprint: 10.26434/chemrxiv.5309668.v3 (2017)

59 Putin, E. et al., Adversarial Threshold Neural Computer for Molecular de Novo Design. *Mol Pharm*, **15** (10), pp. 4386-4397 (2018)

60 Putin, E. et al., Reinforced Adversarial Neural Computer for de Novo Molecular Design. *J Chem Inf Model*, **58** (6), 1194-1204 (2018)

61 De Cao, N. and Kipf, T., MolGAN: An implicit generative model for small molecular graphs. preprint: arXiv:1805.11973v1 (2018)

62 Kuzminykh, D. et al., 3D Molecular Representations Based on the Wave Transform for Convolutional Neural Networks. *Mol Pharm*, **15** (10), pp 4378-4385 (Oct 2018)

63 Zhavoronkov, A. et al., Deep learning enables rapid identification of potent DDR1 kinase inhibitors. Nature Biotechnology, **37** (9), 1038–1040 (2019)

64 Ivanenkov, Y. et al., Are We Opening the Door to a New Era of Medicinal Chemistry or Being Collapsed to a Chemical Singularity? J. Med. Chem., https://pubs.acs.org/doi/full/10.1021/acs.jmedchem.9b00004 (2019)

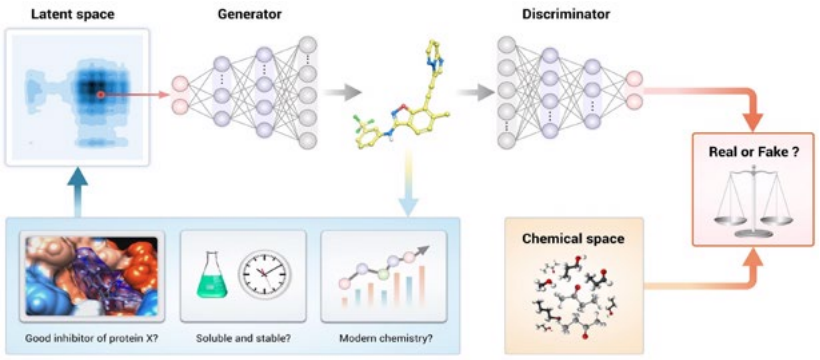

Figure *12*: Pictorial representation of the GANs + RL networks.

At the second stage the model was fine-tuned to expand the latent manifold towards discovering novel inhibitors and preferentially generate DDR1 kinase inhibitors. Reinforcement learning was applied with reward functions based on self-organizing maps (SOM).

The trained GENTRL model generated 30,000 novel unique molecular structures which were filtered with the prioritization pipeline down to 40 molecules selected for synthesis and real-world experiments. Out of six remaining molecules, two had two-digit nanomolar IC50 and high specificity against DDR1 protein.

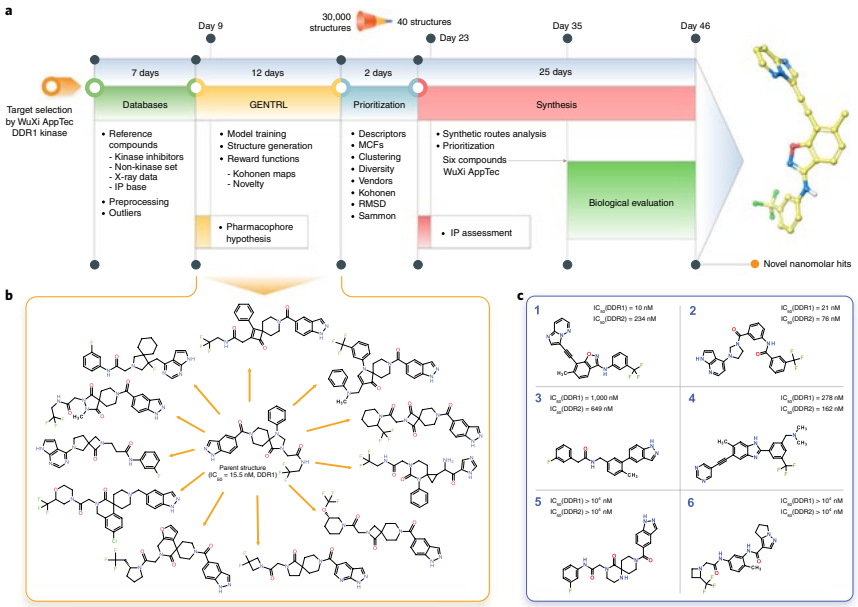

Figure *13*: Timeline and results of the DDR1 project.

Business Results

The generative network can now be used to generate compounds with desired physicochemical properties reducing the need for hand generation of combinatorial libraries of compounds. This approach has the potential to increase the efficiency of lead generation by one or more OOM, and to identify compounds to test against previously hard-to-drug targets across a range of disease areas.

Modeling Side Effects Resulting from Drug Combinations (Polypharmacy)

Background

Many diseases and medical conditions are treated with combinations of drugs. In cancer, for example, combination therapies are being explored to find targeted therapies and biologics to pair with chemotherapy or with one another. Some drugs, for example checkpoint inhibitors (CPIs) like pembrolizumab, have attained blockbuster success, yet fail to improve outcomes for the vast majority of patients. Thus, an entire cottage industry has grown in order to search for the right drugs to combine with CPIs to expand their effectiveness in patients still in need. Another example is the common scenario in which a patient takes multiple drugs as part of their daily regimen and polypharmacy increases the chance of deleterious side effects due to drug-drug interactions.

Adverse events are a major concern in drug development. Every drug must pass stringent toxicity criteria for approval. Yet it would be impractical to test all possible drug pairs for drug-drug induced adverse events. Further, many clinical trials are too small to detect rare but serious polypharmacy effects. Over 15% of the U.S. population is impacted by polypharmacy, and treating the unexpected consequences costs in the hundreds of billions of US dollars.

Case-Study

Approach

Marinka Zitnik and colleagues[65] at Stanford developed a Graph Convolutional Network (GCN) approach to modeling drug-drug interactions, and to predict the specific side effects and complications associated with that interaction. Their model incorporates information from drug-protein (target) and protein-protein interactions, as these relationships were observed to be meaningful in the prevalence of multi-drug prescriptions. A key innovation of the model was not only to predict whether to expect a drug-drug interaction, or polypharmacy side effect, but specifically to identify the type of interaction to expect from among nearly 1,000 defined adverse side effects.

Solution

The authors present *Decagon*, the solution to a "multi-relational link prediction problem in a two-layer multimodal graph/network of two node types: drugs and proteins." Datasets were assembled from published sources: various protein-protein interaction networks; STITCH database for drug-protein interactions; and SIDER, OFFSIDES and TWOSIDES databases for drug-drug interactions. Once the authors harmonized vocabularies, the resulting network included 645 drug and 19,085 protein nodes, linked by 715,612 protein-protein, 18,596 drug-protein and 4,651,131 drug-drug edges.

---

65 Zitnik, M., Agrawal, A., and Leskovec, J. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* **34**, pp i457-i466 (2018)

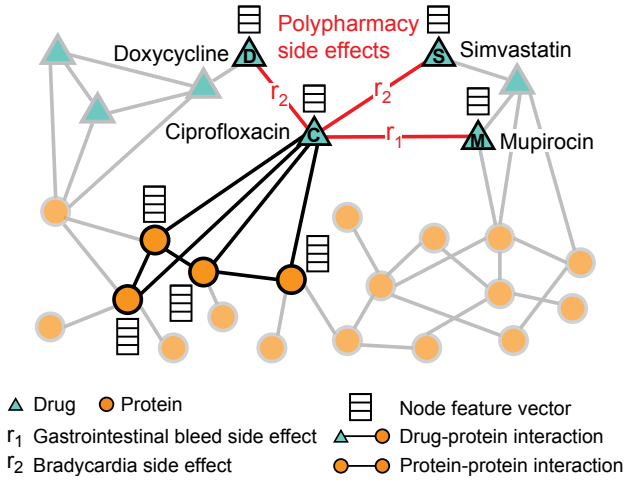

Figure *14*: Illustration of the nodes and edges comprising the protein and drug graph operated on by Decagon.

Decagon consists of an encoder and decoder. The encoder is a GCN that operates on the graph of protein/drug nodes and edges creating embeddings for the nodes. The decoder employs tensor factorization to translate the node embeddings into edges that predict the likelihood and type of polypharmacy events (drug-drug interactions). During training, model parameters are optimized by cross-entropy loss, with 80% of the data used for training and the rest removed for parameter selection and testing/validation. *Decagon* out-performed other tensor factorization and neural embedding approaches to polypharmacy prediction by a wide margin. Its average AUROC score across all 964 side effect types was 0.872, with the nearest comparator at 0.793. It performed especially well for side effects with strong molecular basis.

Business Results

Decagon could be used in the primary care setting as a decision support tool for physicians to guide prescriptions for patients taking several drugs. Alternatively, such a model could be helpful in designing drug combination clinical trials. Eventually, a model of this type might also serve a regulatory function such as contraindication labeling.

THE PROMISE OF AI IN HEALTHCARE

ML models have the ability to take in and train on more data than any one person. ML models can operate at speeds and scales well beyond human capability. ML algorithms can build far more complex models than any human could. Finally, ML is data-driven, and the models can be applied to any narrow problem given proper training data. Due to the complexity of Biology, the rate at which new knowledge is being generated in healthcare, and the reac-

tion speed needed in time critical decision making, ML is a tool that can enable scientists, clinicians, and all medical professionals along the spectrum of biomedical discovery, clinical development, patient care, and population health to make better decisions.

ML helps to reduce failure rates and lower drug development costs by increasing the number and quality of available targets, designing and testing fewer molecules that are more effective at treating disease with limited toxicity or adverse events, and selecting the right patients at the right time for the right treatment in clinical trials. In clinical care, ML drives efficiencies in the clinical workflow itself. Further, ML plays a significant role in aiding decision-making among health practitioners along the continuum of prevention, diagnosis, treatment and patient follow-up. ML permits early, accurate diagnosis by aggregating disparate pieces of information, extracting key patterns from datasets to help identify effective interventions early on when conditions are amenable to treatment.

CONCLUSION

Given the potential benefits of AI in healthcare, but also the real possibility to cause harm, we call for a concerted and collaborative effort to improve industry-wide understanding of the complexities of AI. Critically, the healthcare industry should work together with governments and patients to advance the discussion of responsible AI use. We must work to develop standards that will ensure trustworthiness and transparency in decisions supported by AI as we promote the use of AI in all aspects of healthcare to ensure the best possible decisions are always made.

FUTURE WORK

The current work has just begun to scratch the surface of AI, its use in Healthcare, and the topics that surround it. Future planned papers include:

- A series of works on standards development for AI in healthcare
- Fairness and Trustworthiness as it relates to AI in healthcare
- What an AI enabled IND and NDA might look like
- How AI will change the healthcare industry
- Data and data collection standards

ACKNOWLEDGEMENTS

The authors are grateful to a diverse set of reviewers representing investors, professional writers, health and AI educators, practitioners and researchers that helped to improve this work. In particular we wish to thank:

- Brian Demmert – Founding Partner, Life Science & Healthcare at Armentum Partners
- Johan Van Helleputte – Former SVP Strategic Planning at IMEC
- Paul Agepow – Director Health Informatics at Astrazeneca
- Susan Harvey – VP Global Medical Affairs at Hologic, Breast and Skeletal
- Rick Starrs – CEO at the National Association of Veterans Research & Education Foundation (NAVREF)
- Lee Lendenberger – Writer at BioWorld
- Robert C Bollinger – Professor of Medicine, Public Health and Nursing at Johns Hopkins Medicine, and Director of Center for Clinical Global Health Education
- Nayoung Louie – Lecturer at Johns Hopkins Carey Business School

Appendix:

DEEPER TECHNICAL DETAILS

Train / Test Assumption

Machine learning is primarily concerned with accurate predictions, therefore, the objective is to build a model that performs the best on the test set. The way this is achieved is by having the machine learning algorithm construct a mathematical function (which is the model) that minimizes the prediction error. The test set is often unknown, and thus the labels on the test set are unknown. The model is therefore generated by minimizing a loss function over a training set, requiring a key assumption. **The assumption is that the training set and the test set are both derived from the same underlying statistical distribution.** In other words, the patterns found in the training set are assumed to be the same patterns found in the test set. This seems obvious, but ignorance of, or disregard for, this assumption leads to the vast majority of the mistakes and misapplications of supervised machine learning models. Discord between the distribution of training and test set data is a major source of bias. Unfortunately, models will always contain some bias because a finite training set will never be completely representative of all other possible test sets. Significantly, models can still be applied when careful characterization, awareness, and reduction of the biases is performed. One example of bias is a model built to predict cancer risk using young people in a training set and applying the model to a much older population.

Model Performance

The specific metric to evaluate model performance depends on the model function, loss function, data balance, and the final intent of the model. While the term accuracy is often used interchangeably with the term performance, this is not always correct. For example, accuracy in the case of binary classification models refers to the proportion of correct predictions made by the model on the test sets. When discussing model performance, sticking to well-defined standard terms will help to reduce confusion.

In addition to using the proper term one must also be very specific as to what dataset the metric is referring. The best practice when testing supervised ML model performance is to use a labeled dataset that was not part of the training set: the **holdout dataset.** A holdout dataset is used to provide a better predictor of model performance on a prospective test dataset. How the holdout dataset is constructed relative to the training dataset is critical for getting a proper measure of prospective model performance. Generally, the more dissimilar the holdout dataset is to the training dataset the more the metric will provide a measurement of the model's generalizability (discussed in more detail below).

For binary classifications such as diagnostics, performance is generally described in terms of a confusion matrix (Table 1), which charts:

- true positives (TP), number of **correct positive** predictions;
- true negatives (TN), number of **correct negative** predictions;
- false positives (FP), number of **incorrect positive** predictions;
- false negatives (FN), number of **incorrect negative** predictions.

Table *1*: Confusion Matrix template

Test Results	Disease Present	Disease Absent
Positive	TP	FP
Negative	FN	TN

From these values, additional metrics are defined to assist in interpreting performance. **Sensitivity**, also known as recall, or true positive rate, represents the ratio of correctly identified positives over the entire dataset (TP / TP + FN). Meanwhile, **specificity** describes the ratio of correctly identified negatives over the entire dataset (TN / TN + FP). Most classification algorithms can balance between model specificity and sensitivity through changing the discrimination threshold. Coupled, these two metrics provide a human-interpretable evaluation of model performance, but can be problematic for model benchmarking if one has favorable sensitivity and the other has favorable specificity. Thus, model performance is typically compared by plotting the relationship between true positive rates and false positives rates of different models, taken at varied discrimination thresholds. This relationship is known as receiver operating characteristic (ROC) curve, and often summarized with a single value, Area Under the Curve (AUC), which ranges from 0.5 (no predictive power) to 1 (perfect predictions). A variant known as Precision-Recall Area Under the Curve (PR-AUC), substitutes precision (TP / TP + FP) for false positive rate, which does not account for true negatives and can provide a better description of performance with highly imbalanced datasets. PR-AUCs are also appropriate for ranking tasks.

For regression tasks, performance evaluation generally involves some measure of distance between the predicted value and the true value for each data point in the test sets. The MSE is a common metric to summarize predictive performance across a set of testing examples. R-squared is another metric that is often used. R-squared relates the MSE of the model to the MSE of a null model. A null model is a model that always predicts the same value, namely the average value of the training set labels.[66]

For ranking tasks, metrics that measure the relative ordering are used to evaluate model performance. The dominant metrics are Kendall's tau (τ) and Spearman's rank correlation

66 Alexander, D. et al., Beware of R2: simple, unambiguous assessment of the prediction accuracy of QSAR and QSPR models. *J. Chem. Inf. Model.,* **55** (7), pp. 1316-1322 (2015)

coefficient (ρ). Kendell's tau counts and compares concordant and discordant pairs in the sample, which directly relates to the underlying ranking. Because the metric is based purely on pair counts, the algorithm generating the pairs can account for dataset characteristics, such as noise. Spearman's rank correlation coefficient measures the covariance of the sample. Unlike Kendell's tau, Spearman's rank correlation coefficient can be affected by noise in the data.

Applying these performance metrics to test and holdout sets provides a way to compare models generated by different algorithms or parameter sets. Ultimately however, evaluating the true performance of a supervised machine learning model requires repeat testing in a real-world setting. Ideally, real-world performance should reflect the performance in the model testing stages using a proper holdout set. A drop in real-world performance could be attributed to many factors, including overfitting, underfitting, non-representative training/testing data, or testing data that are too similar to examples in the training set (memorization). Models that demonstrate consistent performance across a large range of applications and inputs are said to be generalizable.

Overfitting and Bias

Supervised learning typically experiences a tradeoff between a model's ability to identify relevant predictive relationships and generalizability (Figure 15). The use of too few data features or a simpler predictive function creates models with low complexity. This can compromise the identification of predictive relationships in a process known as *underfitting*. These models are said to have high *bi*as. In contrast, *overfitting* stems from the predictive algorithm learning patterns from random noise, often by over-minimizing the loss function. This leads to the algorithm choosing a model function that is too complex and these models are said to have high *variance*. The ideal model development seeks to strike a balance between bias and variance to produce effective, generalizable models.

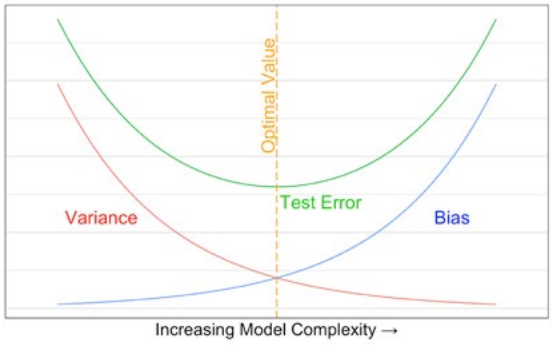

Figure *15*: Optimal model complexity strikes a balance between variance and bias.

To further illustrate the concept of overfitting (Figure 16), training data were generated by adding Gaussian noise to random data points from a linear equation. Without knowing how the data were generated, one may intuit by looking at the plot that these data should be linear fit. However, naïvely a higher order polynomial would actually further minimize the

training set error. Ultimately, a polynomial where the number of parameters equals the number of data points results in training set error of zero, but also will result in a nonfunctional model. In a more realistic higher dimensional case, it is generally not clear where the optimal balance lies.

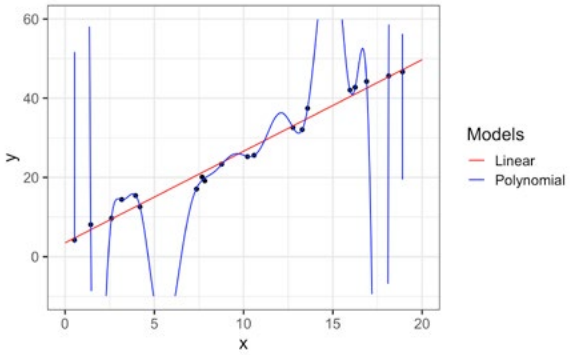

Figure *16*: Two models (linear and polynomial) fit to noisy data. The MSE training loss for the linear and polynomial model are 82.2 and 1.1 respectively.

A number of techniques may be employed to avoid overfitting. One way to avoid this issue is to use cross-fold validation—a set of models are built on multiple subsets of the training data, while evaluating the loss on the remaining training data (the validation set), and then selecting the model with the best performance on the validation data. Another way is by using a technique called regularization, which is often built into supervised machine learning algorithms. Regularization is an approach that pushes the machine learning algorithm to choose the simplest function to fit the data thereby helping to avoid overfitting.

To further illustrate the concept of underfitting and bias (Figure 17), data were generated by adding Gaussian noise to random points along a third order polynomial curve. If only the data between -5 and 5 were available, the model would be very biased and underfit outside of that range. This demonstrates the issue of applying a model outside of the domain of the training data. If the machine learning algorithm is unduly restricted to a linear function, the final model will also be underfit, but in this case the final loss should indicate a problem by being very high. Finally, a third order polynomial fit strikes the balance between bias and variance.

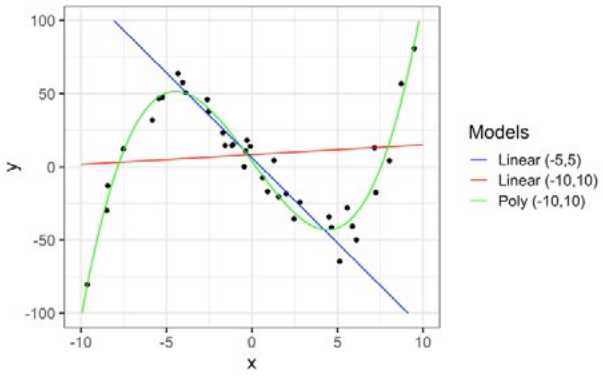

Figure *17*: Two linear fits to different ranges of the data and one third order polynomial fit.