EDITOR: Irena Bojanova, NIST, irena.bojanova@computer.org

How Much to Trust Artificial Intelligence?

George Hurlburt, STEMCorp

There has been a great deal of recent buzz about the rather dated notion of artificial intelligence (AI). AI surrounds us, involving numerous applications ranging from Google search, to Uber or Lyft ride-summoning, to airline pricing, to Alexa or Siri. To some, AI is a form of salvation, ultimately improving quality of life while infusing innovation across myriad established industries. Others, however, sound dire warnings that we will all soon be totally subjugated to superior machine intelligence. AI is typically, but no longer always, software dominant, and software is prone to vulnerabilities. Given this, how do we know that the AI itself is sufficiently reliable to do its job, or—put more succinctly—how much should we trust the outcomes generated by AI?

Risks of Misplaced Trust

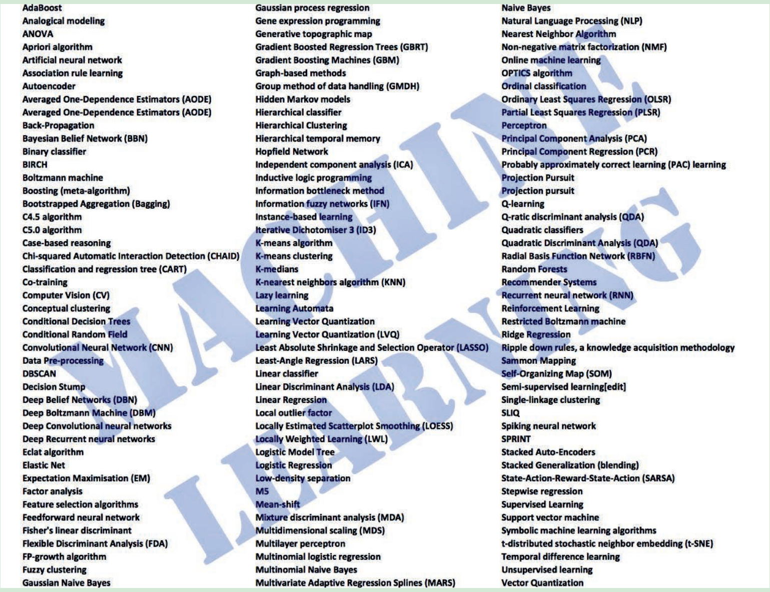
Consider the case of self-driving cars. Elements of AI come into play in growing numbers of self-driving car autopilot regimes. This results in vehicles that obey the rules of the road, except when they do not. Such was the case when a motor vehicle in autonomous mode broadsided a turning truck in Florida, killing its "driver." The accident was ultimately attributed to driver error, as the autonomous controls were deemed to be performing within their design envelope. The avoidance system design at the time required that the radar and visual systems agree before evasive action would be engaged. Evidence suggests, however, that the visual system encountered glare from the white truck turning against bright sunlight. This system neither perceived nor responded to the looming hazard. At impact, however, other evidence implicated the "driver," who was watching a *Harry Potter* movie. The driver, evidently overconfident of the autopilot, did not actively monitor its behavior and failed to override it, despite an estimated seven-second visible risk of collision.[1] The design assurance level was established, but the driver failed to appreciate that his autopilot still required his full, undivided attention. In this rare case, misplaced trust in an AI-based system turned deadly.

Establishing a Bar for Trust

AI advancement is indeed impressive. DARPA, sponsor of early successful autonomous vehicle competitions, completed the Cyber Grand Challenge (CGC) competition in late 2016. The CGC established that machines, acting alone, could play an established live hacker's game known as Capture the Flag. Here, a "flag" is hidden in code, and the hacker's job is to exploit vulnerabilities to reach and compromise an opponent's flag. The CGC offered a $2 million prize to the winning team that most successfully competed in the game. The final CGC round pitted seven machines against one another on a common closed network without any human intervention. The machines had to identify vulnerabilities in an opponent's system, fix them on their own system, and exploit them in

1520-9202/17/$33.00 © 2017 IEEE

opponents' systems to capture the flag. Team Mayhem from Carnegie Mellon University was declared the winner.[2]



MACHINE
LEARNING

AdaBoost
Analogical modeling
ANOVA
Apriori algorithm
Artificial neural network
Association rule learning
Autoencoder
Averaged One-Dependence Estimators (AODE)
Averaged One-Dependence Estimators (AODE)
Back-Propagation
Bayesian Belief Network (BBN)
Binary classifier
BIRCH
Boltzmann machine
Boosting (meta-algorithm)
Bootstrapped Aggregation (Bagging)
C4.5 algorithm
C5.0 algorithm
Case-based reasoning
Chi-squared Automatic Interaction Detection (CHAID)
Classification and regression tree (CART)
Co-training
Computer Vision (CV)
Conceptual clustering
Conditional Decision Trees
Conditional Random Field
Convolutional Neural Network (CNN)
Data Pre-processing
DBSCAN
Decision Stump
Deep Belief Networks (DBN)
Deep Boltzmann Machine (DBM)
Deep Convolutional neural networks
Deep Recurrent neural networks
Eclat algorithm
Elastic Net
Expectation Maximisation (EM)
Factor analysis
Feature selection algorithms
Feedforward neural network
Fisher's linear discriminant
Flexible Discriminant Analysis (FDA)
FP-growth algorithm
Fuzzy clustering
Gaussian Naive Bayes
Gaussian process regression
Gene expression programming
Generative topographic map
Gradient Boosted Regression Trees (GBRT)
Gradient Boosting Machines (GBM)
Graph-based methods
Group method of data handling (GMDH)
Hidden Markov models
Hierarchical classifier
Hierarchical Clustering
Hierarchical temporal memory
Hopfield Network
Independent component analysis (ICA)
Inductive logic programming
Information bottleneck method
Information fuzzy networks (IFN)
Instance-based learning
Iterative Dichotomiser 3 (ID3)
K-means algorithm
K-means clustering
K-medians
K-nearest neighbors algorithm (KNN)
Lazy learning
Learning Automata
Learning Vector Quantization
Learning Vector Quantization (LVQ)
Least Absolute Shrinkage and Selection Operator (LASSO)
Least-Angle Regression (LARS)
Linear classifier
Linear Discriminant Analysis (LDA)
Linear Regression
Local outlier factor
Locally Estimated Scatterplot Smoothing (LOESS)
Locally Weighted Learning (LWL)
Logistic Model Tree
Logistic Regression
Low-density separation
M5
Mean-shift
Mixture discriminant analysis (MDA)
Multidimensional scaling (MDS)
Multilayer perceptron
Multinomial logistic regression
Multinomial Naive Bayes
Multivariate Adaptive Regression Splines (MARS)
Naive Bayes
Natural Language Processing (NLP)
Nearest Neighbor Algorithm
Non-negative matrix factorization (NMF)
Online machine learning
OPTICS algorithm
Ordinal classification
Ordinary Least Squares Regression (OLSR)
Partial Least Squares Regression (PLSR)
Perceptron
Principal Component Analysis (PCA)
Principal Component Regression (PCR)
Probably approximately correct learning (PAC) learning
Projection Pursuit
Projection pursuit
Q-learning
Q-ratic discriminant analysis (QDA)
Quadratic classifiers
Quadratic Discriminant Analysis (QDA)
Radial Basis Function Network (RBFN)
Random Forests
Recommender Systems
Recurrent neural network (RNN)
Reinforcement Learning
Restricted Boltzmann machine
Ridge Regression
Ripple down rules, a knowledge acquisition methodology
Sammon Mapping
Self-Organizing Map (SOM)
Semi-supervised learning[edit]
Single-linkage clustering
SLIQ
Spiking neural network
SPRINT
Stacked Auto-Encoders
Stacked Generalization (blending)
State-Action-Reward-State-Action (SARSA)
Stepwise regression
Supervised Learning
Support vector machine
Symbolic machine learning algorithms
t-distributed stochastic neighbor embedding (t-SNE)
Temporal difference learning
Unsupervised learning
Vector Quantization

Figure 1. Some prevalent AI machine learning algorithms.

John Launchbury, director of DARPA's Information Innovation Office, characterizes the type of AI associated with the CGC as *handcrafted knowledge*. Emerging from early expert systems, this technology remains vital to the advancement of modern AI. In handcrafted knowledge, systems reason against elaborate, manually defined rule sets. This type of AI has strength in reasoning but is limited in forms of perception. However, it possesses no ability to learn or perform abstraction.[3]

While building confidence that future reasoning AI can indeed rapidly diagnose and repair software vulnerabilities, it is important to note that the CGC was intentionally limited in scope. The open source operating system extension was simplified for purposes of the competition,[4] and known malware instances were implanted as watered-down versions of their real-life counterparts.[5] This intentionally eased the development burden, permitted a uniform basis for competitive evaluation, and reduced the risk of releasing competitors' software into the larger networked world without requiring significant modification.

The use of "dirty tricks" to defeat an opponent in the game adds yet another, darker dimension. Although the ability to re-engineer code to rapidly isolate and fix vulnerabilities is good, it is quite another thing to turn these vulnerabilities into opportunities that efficiently exploit other code. Some fear that if such a capability were to be unleashed and grow out of control, it could become a form of "supercode"—both exempt from common vulnerabilities and capable of harnessing the same vulnerabilities to assume control over others' networks, including the growing and potentially vulnerable Internet of Things (IoT). This concern prompted the Electronic Frontier Foundation to call for a "moral code" among AI developers to limit reasoning systems to perform in a trustworthy fashion.[4]

Machine Learning Ups the Trust Ante

Launchbury ascribes the term *statistical learning* to what he deems the second wave of AI. Here, perception and learning are strong, but the technology lacks any ability to perform reasoning and abstraction. While statistically impressive, machine learning periodically produces individually unreliable results, often manifesting as bizarre outliers. Machine learning can also be skewed over time by tainted training data.[3] Given that not all AI learning yields predictable outcomes, leading to the reality that AI systems could go awry in unexpected ways, effectively defining the level of trust in AI based tools becomes a high hurdle.[6]

At its core, AI is a high-order construct. In practice, numerous loosely federated practices and algorithms appear to compose most AI instances—often crossing many topical domains. Indeed, AI extends well beyond computer science to include domains such as neuroscience, linguistics, mathematics, statistics, physics, psychology, physiology, network science, ethics, and many others. Figure 1 depicts a less than fully inclusive list of algorithms that underlie second-wave AI phenomena, often collectively known as machine learning.

This myriad of potential underlying algorithms and methods available to achieve some state of machine learning raises some significant trust issues, especially for those involved in software testing as an established means to assure trust. When the AI becomes associated with mission criticality, as is increasingly the case, the tester must establish the basis for multiple factors, such as programmatic consistency, repeatability, penetrability, applied path tracing, or identifiable systemic failure modes.

The nontrivial question of what is the most appropriate AI algorithm goes as far back as 1976.[3] The everyday AI practitioner faces perplexing issues regarding which is the right algorithm to use to suit the desired AI design. Given an intended outcome, which algorithm is the most accurate? Which is the most efficient? Which is the most straightforward to implement in the anticipated environment? Which one holds the greatest potential for the least corruption over time? Which ones are the most familiar and thus the most likely to be engaged? Is the design based on some form of centrality, distributed agents, or even swarming software agency? How is this all to be tested?

The everyday AI practitioner faces perplexing issues regarding which is the right algorithm to use to suit the desired AI design.

These questions suggest that necessary design tradeoffs exist between a wide range of alternative AI-related algorithms and techniques. The fact that such alternative approaches to AI exist at all suggests that most AI architectures are far from consistent or cohesive. Worse, a high degree of contextually-based customization is required for both reasoning and learning systems. This, of course, extends to AI testing, because each algorithm and its custom implementation brings its own unique deep testing challenges, even at the unit level.

One high-level AI test assesses the ability to correctly recognize and classify an image. In some instances, this test has surpassed human capability to make such assessments. For example, the Labeled Faces in the Wild (LFW) dataset supports facial recognition with some 13,000 images to train and calibrate facial recognition machine learning tools using either neural nets or deep learning. The new automated AI image recognition tools can statistically outperform human facial recognition capability using this dataset.[7] The task at hand, however, is fundamentally perceptual in nature. These tasks functionally discriminate through mathematically correlated geometric patterns but stop short of any form of higher-order cognitive reasoning. Moreover, while it compares selective recognition accuracy against human ability, other mission-critical aspects of the underlying code base remain unchecked under this test.

Beyond the Code

Testing machine learning becomes further complicated as extensive datasets are required to "train" the AI in a learning environment. Not only should the AI code be shown to be flawless, but the data used in the training should theoretically bear the highest pedigree. In the real world, however, datasets often tend to be unbalanced, sparse, inconsistent, and often inaccurate, if not totally corrupt. Figure 2 suggests that information often results from resolving ambiguity. Even under controlled conditions, significant differences result between the use of single or multiple well-validated datasets used to train and test classifiers. Thus, even controlled testing for classifiers can become highly complicated and must be approached carefully.[8]

Other trust-related factors extend well beyond code. Because coding is simultaneously a creative act and somewhat of a syntactic science, it is subject to some degree of interpretation. It is feasible that a coder can inject either intentional or unintentional cultural or personal bias into the resulting AI code. Consider the case of the coder who creates a highly accurate facial recognition routine but neglects to consider skin pigmentation as a deciding factor among the recognition criteria. This action could skew the results away from features otherwise reinforced by skin color. Conversely, the rates of recidivism among criminals skews some AI-based prison release decisions along racial lines. This means that some incarcerated individuals stand a better statistical chance of gaining early release than others—regardless of prevailing circumstances.[9] Semantic inconsistency can further jeopardize the neutrality of AI code, especially if natural language processing or idiomatic speech recognition are involved.

Some suggest that all IT careers are now cybersecurity careers.[10] This too has a huge implication for the field of AI development and its implementation. The question of "who knows what the machine knew and when it knew it" becomes significant from a cybersecurity standpoint. What a machine learns is often not readily observable, but rather lies deeply encoded. This not only affects newly internalized data, but—in the IoT—these data can trip decision triggers to enliven actuators that translate the "learning" into some sort of action. Lacking concrete stimulus identity and pedigree, the overall AI-sparked IoT stimulus-response mechanism becomes equally uncertain. Nonetheless, the resulting actions in mission-critical systems require rigorous validation.

The Third Wave

Launchbury foresees the need for a yet-to-be-perfected third wave of AI, which he names *contextual adaptation*. This technology, requiring much more work, brings together strengths in perception, learning, and reasoning and supports a significantly heightened level of cross-domain abstraction.[3]

The 2017 Ontology Summit, aptly entitled "AI, Learning, Reasoning, and Ontologies," concluded in May 2017. Reinforcing Launchbury's observation, the draft summit communique concluded that, to date, most AI approaches, including machine learning tools, operate at a subsymbolic level using computational techniques that do not approximate human thought. Although great progress has been achieved in many forms of AI, the full treatment of knowledge representation at the symbolic level awaits maturity (bit.ly/2qMN0it). Correspondingly, the utility of ontology as a formal semantic organizing tool offers only limited advantages to AI and its ultimate test environment.

The semantic network involves graph representations of knowledge in the form of nodes and arcs. It provides a way to understand and visualize relationships between symbols, often represented by active words, which convey varying meanings when viewed in context. AI, largely subsymbolic today, will need to deal with applied semantics in a far more formal sense to achieve third-wave status. Under such circumstances, AI becomes nonlinear, in which cause and effect are increasingly decoupled via multiple execution threads. This leads to the establishment of *complex adaptive systems* (CAS), which tend to adhere to and be influenced by nonlinear network behavior.

information technology (IT) as "the study
support or management
information
information for decision making
and provide productivity
Association for Computing

Figure 2. Information provenance can often be unclear.

In a CAS, new behaviors emerge based on environmental circumstance over time. Here, there can be multiple self-organizing paths leading to success or failure, all triggered by highly diversified nodes and arcs that can come, grow, shrink, and go over time. Such networks defy traditional recursive unit testing when composed using embedded software, which is interrelated to data. This is because in a CAS, the whole often becomes far more than merely the sum of the parts.[11] Rather, new approaches, emerging from applied network science, offer a better means of assessing dynamic AI behavior that emerges over time. This becomes increasingly true as the temporal metrics associated with graph theory become better understood as a means of describing dynamic behaviors that fail to follow linear paths to achieve some desired effect.[12]

Until some reliable methodology is adopted for the assessment of assured trust within AI, the watchword must be caution. Any tendency to put blind faith in what in effect remains largely untrusted technology can lead to misleading and sometimes dangerous conclusions.

References

1. N.E. Boudette, "Tesla's Self-Driving System Cleared in Deadly Crash," *New York Times*, 19 Jan. 2017.
2. D. Coldewey, "Carnegie Mellon's Mayhem AI Takes Home $2 Million

from DARPA's Cyber Grand Challenge," *TechCrunch*, 5 Aug. 2016; tcrn.ch/2aM3iS7.

3. J. Launchbury, "A DARPA Perspective on Artificial Intelligence," DARPAtv, 15 Feb. 2017; www.youtube.com/watch?v5-O01G3tSYpU.
4. N. Cardozo, P. Eckersley, and J. Gillula, "Does DARPA's Cyber Grand Challenge Need a Safety Protocol?" Electronic Frontier Foundation, 4 Aug. 2016; bit.ly/2aPxRXc.
5. A. Nordrum, "Autonomous Security Bots Seek and Destroy Software Bugs in DARPA Cyber Grand Challenge," *IEEE Spectrum*, Aug. 2016; bit.ly/2arLOcR.
6. S. Jontz, "Cyber Network, Heal Thyself," *Signal,* 1 Apr. 2017; bit.ly/2o0ZCVe.
7. A. Jacob, "Forget the Turing Test—There Are Better Ways of Judging AI," *New Scientist*, 21 Sept. 2015; bit.ly/1MoMUnF.
8. J. Demsar, "Statistical Comparisons of Classifiers over Multiple Data Sets," *J. Machine Learning Research*, vol. 7, 2006, pp. 1–30.
9. H. Reese, "Bias in Machine Learning, and How to Stop It," *TechRepublic*, 18 Nov. 2016; tek.io/2gcqFrI.
10. C. Mims, "All IT Jobs Are Cybersecurity Jobs Now," *Wall Street J.*, 17 May 2017; on.wsj.com/2qH5VP2.
11. P. Erdi, *Complexity Explained*, Springer-Verlag, 2008.
12. N. Masuda and R. Lambiotte, *A Guide to Temporal Networks*, World Scientific Publishing, 2016.

George Hurlburt *is chief scientist at STEMCorp, a nonprofit that works to further economic development via adoption of network science and to advance autonomous technologies as useful tools for human use. He is engaged in dynamic graph-based Internet of Things architecture. Hurlburt is on the editorial board of* IT Professional *and is a member of the board of governors of the Southern Maryland Higher Education Center. Contact him at ghurlburt@change-index.com.*