AI in Cyberspace: Beyond the Hype

Fernando Maymí
Scott Lathrop

Artificial intelligence (AI) is quickly becoming ubiquitous, particularly as part of solutions to defense problems in cyberspace. It seems like few companies want to risk marketing products that cannot be described using this term, perhaps for fear of losing ground to competitors who can. But what exactly is meant by AI? Is it all just marketing hype? The answer, of course, is far from simple. To move beyond the hype, we need to look at what AI is, what it is not and how the technology needs to mature to live up to its promise.

What it is

AI is a multidisciplinary field primarily associated with computer science, with influences from mathematics, cognitive psychology, philosophy, and linguistics (among others). The term was originally coined at a Dartmouth College workshop in 1956 and continues to be characterized by cycles of excitement, marvel, and disappointment as we come to grips with and gain a better understanding of both its promises and limitations. Depending on who you ask, AI's goals range from creating general intelligent systems to modeling human cognitive processes, to achieving superhuman performance on very specific tasks. An example of this is what we are beginning to see in image recognition systems through a machine learning technique called deep learning (more on that later). For this article, we are focused on defining AI in terms of how it can improve the functionality of a system so that certain tasks require decreased human involvement and intervention.

Fernando Maymí, Ph.D., CISSP, is Lead Scientist in the Cyber and Secure Autonomy Division of Soar Technology, Inc., an artificial intelligence research and development company where he leads multiple advanced research projects developing autonomous cyberspace agents for the Department of Defense. Dr. Maymi is a retired Army Officer with more than 25 years of service; he was the first Deputy Director of the Army Cyber Institute at West Point, an organization he helped grow and lead from its inception, and a former West Point faculty member. Dr. Maymí holds three patents and regularly consults on cybersecurity issues both in the U.S. and abroad. He is the author of numerous publications including the best-selling *CISSP All-in-One Exam Guide*.

From a historical perspective, what is considered AI today may not be considered "intelligent" or "cutting-edge" tomorrow. In the 1980s, a grammar checker seemed intelligent though such algorithms are now just part of word processing software. When web search started, people were amazed at search engines such as Google. Voice recognition is now integrated into our daily lives through technology such as Amazon's Alexa™ and Apple's Siri™; these AI technologies seemed "intelligent" when they first arrived on the scene but are now simply part of our lives. In the future, the same will be true for driverless cars, and other AI adopted technology.

At a high level, AI can be divided into two different approaches as shown in Figure 1 [1]: symbolic and non-symbolic; the key difference is in how each represents knowledge. Both approaches are concerned with how knowledge is organized, how inference proceeds to support decision-making, and how the system learns. For example, a spam filter may organize knowledge about an email message as a vector of words. The system learns as it is trained on whether messages are spam or benign. This training adjusts the system's internal knowledge model. After training, each time a new email message arrives, the trained system infers whether the message is spam by comparing its features to the system's underlying knowledge model.

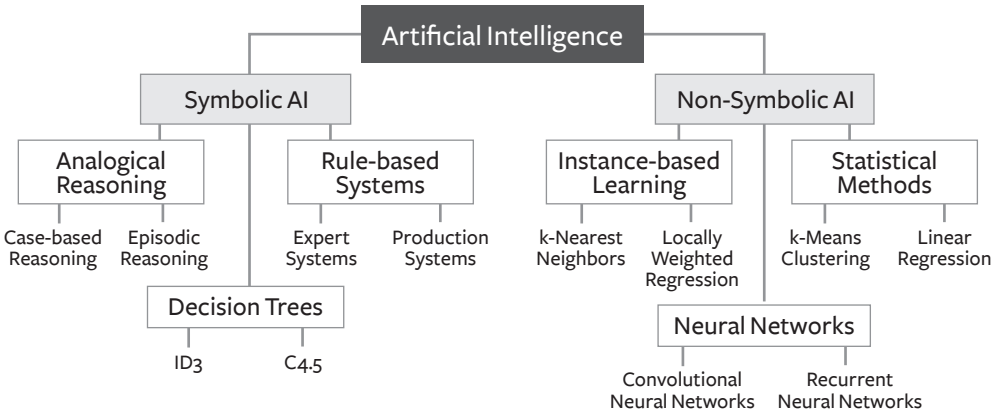

Figure 1: A partial taxonomy of artificial intelligence

Scott Lathrop, Ph.D., CISSP, is Soar Technology's Director of Cyber and Secure Autonomy. Before joining SoarTech, Dr. Lathrop served 26 years as an officer in the Army with his last military assignment as the Director of Research & Development at the United States Cyber Command where he led the delivery of multi-million-dollar full spectrum, technically assured, cyberspace capabilities as the Chief Technology Officer to Commander/Director USCYBERCOM/NSA. Prior to USCYBERCOM, Dr. Lathrop served as an Associate Professor in the Department of Electrical Engineering and Computer Science at the United States Military Academy, designing West Point's initial cybersecurity program, helping stand up the robotics program, and directing the Artificial Intelligence courses. Dr. Lathrop is a renowned author in the cognitive architecture community, publishing on mental imagery and applying cognitive architectures to robotics and cyber-entities.

Symbolic AI

In symbolic approaches to AI, system developers model real-world concepts, their relationships, and how they interact to solve a set of problems using a set of symbols (e.g., words or tokens). These AI approaches commonly use ontologies to organize knowledge and heuristic-based rules to support reasoning. Symbolic systems may also learn, such as learning a decision tree based on provided examples or through learning an appropriate decision based on previously recorded, similar events. Symbolic AI requires considerable *knowledge engineering* of both the problem and solution domains, which makes it fairly labor-intensive. However, it yields results that are inherently explainable to humans since they are derived from human knowledge models in the first place. Symbolic AI systems include the expert systems that became prolific in the 1980s. These relied on extensive interviewing of subject matter experts and time-consuming encoding of their expertise in a series of conditional structures. Unsurprisingly, these early systems were unable to adapt or learn absent human intervention, which is a problem when we consider the number of exceptions that apply to almost all processes.

The systems developed as part of the DARPA Cyber Grand Challenge are primarily symbolic AI systems. These automated reasoners can identify vulnerabilities in software services, develop a patch, and deploy the patch at machine speed. To create these systems, the teams encoded the knowledge associated with the types of vulnerabilities they might find (a form of an ontology), the procedures for finding the vulnerabilities (search and reasoning), and possible methods to remediate those vulnerabilities (decision-making). The systems learned in the sense that they were able to explore their environment (i.e. the network that they were a part of) and identify vulnerable services. However, that learning did not include finding new

vulnerabilities for which they were not previously encoded to identify. Furthermore, the systems did not learn how to identify new vulnerabilities by being trained on previous vulnerabilities through offline learning (i.e., learning from data before the system is deployed), which is common in the non-symbolic, statistical machine learning approaches discussed below. Nonetheless, these systems generated impressive results and will prove useful as we continue to investigate ways to make cyber defense more autonomous.

Modern symbolic systems are exemplified by cognitive architectures that emulate the way in which our human brains work. Systems like Carnegie Mellon University's Adaptive Control of Thought - Rational (ACT-R) and the University of Michigan's Soar (both open-source projects) are commonly used to build AI systems that can solve large sets of complex, real-world problems. Like their early symbolic predecessors, ACT-R and Soar require a fair amount of knowledge engineering in the form of building cognitive models to bootstrap them. Unlike early systems, however, these newer cognitive frameworks are capable of learning through interactions with their environments without human assistance and incorporate non-symbolic, machine learning approaches as part of their architectures. These "co-symbolic" (i.e. a hybrid symbolic/non-symbolic system) approaches appear to be where AI is heading as it takes advantage of the non-symbolic learning with the explainability of symbolic systems.

Non-symbolic AI

Another approach to AI departs from the use of symbolic representations of human knowledge and focuses instead on learning patterns in data for classifying objects, predicting future results, or clustering similar sets of data. Non-symbolic AI approaches are where many of the most recent advances have occurred, primarily in classification tasks such as image and voice recognition. In the current vernacular, these non-symbolic approaches are commonly called machine learning (ML) even though, as we just discussed, symbolic systems may also learn. As with symbolic approaches, non-symbolic ML systems also incorporate knowledge representations and reasoning. The knowledge representation is typically quantitative vectors (i.e., non-symbolic) with features from the dataset that describe the input (e.g., the pixels from an image, frequencies from an audio file, word vectors). Whereas symbolic AI requires considerable knowledge engineering, non-symbolic AI generally requires significant *data acquisition and data curating*, which can be labor-intensive even for domains where data is readily available. However, rather than having to program the knowledge as in a symbolic system, the non-symbolic ML system learns its knowledge, in the form of numeric parameters (i.e., weights), through offline [2] training with datasets with millions of examples. The most successful non-symbolic ML approaches today are *supervised learning*, where the datasets include a label or the "answer" for the correct classification. As training progresses, the ML model learns the correct parameters (i.e., weights) that minimize a cost function enabling the match of input patterns to an output classification or prediction. Reasoning then occurs when the trained ML model receives input from the operational environment and infers a classification.

Classification determines the class of a new sample based on what is known about previous samples. A common example of this is an algorithm called k-Nearest Neighbors (KNN), which is a supervised learning technique in which the nearest *k* neighbors influence the classification of the new point (e.g., if more than half of its *k* nearest neighbors are in one class, then the new point also belongs in that class). For cybersecurity, this is helpful when trying to determine whether a binary file is malware or detecting whether an email is spam.

Prediction compares previous data samples and determines what the next sample(s) should be. If you ever took a statistics class in college, you may recall a type of analysis called regression, in which you try to determine the line (or curve) that most closely approximates a sequence of data points. We use the same approach to prediction in ML by learning from previous observations to determine where the next data point(s) should appear, which is useful for network flow analysis.

In clustering, or *unsupervised learning*, on the other hand, we do not have a preconception of which classes (or even how many) exist; we determine where the samples naturally clump together. One of the most frequently-used clustering algorithms is k-Means clustering, in which new data points are added to one of the k clusters based on which one is closest to the new point [3]. Clustering is useful for anomaly detection.

Finally, *reinforcement learning* tunes decision-making parameters towards choices that lead to positive outcomes in the environment. For example, one might have a security analyst provide feedback to an anomaly detector when it incorrectly classifies a benign alert as malicious (i.e., false positive). This feedback adjusts the internal model's weights so that the anomaly classification improves.

ML can be divided into two schools of thought. The first school tries to model the physiology of the brain and, specifically, the roles of neurons and synapses. This school gave rise to artificial neural networks, which break down complex problems into a multitude of tiny problems. For example, the problem of finding a face in a photograph is commonly broken down into problems such as deciding whether an eye, nose, and ear are in the frame and whether they are in the correct locations relative to each other. The "connection" between neurons is a simple mathematical function so that the output of the first neuron (e.g., there is an eye in the frame) is fed into the input of the next connected neuron by a multiplicative parameter that determines the weight of the connection. These parameters, or weights, are what are adjusted through algorithms such as backpropagation that enable the system to learn to match the input pattern to the desired output classification or prediction.

Neural networks are assembled into layers so that neurons in the same layer seldom pass data to each other and, instead, pass it to the next layer. The more layers you have, the more complex the problem you can classify or predict (e.g., the difference between classifying handwritten digits versus classifying dogs and cats in an image). A neural network with

many layers [4] is considered capable of deep learning. A fairly deep neural network will require significantly more computing resources and training data than its "shallow" brethren. Therefore, depending on the problem at hand, deep learning may be an undesirable overkill.

The second school of thought in ML dispenses with any attempt to model physiology and focuses instead on mathematical algorithms that exploit anything from Euclidian distance to statistical regression to probabilistic (e.g., Bayesian) methods. Regardless of to which school it belongs, all ML is focused on specific features of the data (e.g., source IP address, interarrival rate). Given enough prior data, we can usually find good ways to classify, predict, or cluster new observations. The catch is that many, if not most, cybersecurity applications, require labeled (or at least partially labeled) data sets that represent the statistical distribution of the data in the operational environment. This means that if we want to train a supervised ML system to recognize malicious traffic, we need that traffic to be labeled as such and it must be representative of the number of malicious samples we would see in the real world. Acquiring these realistic and sufficiently large sets of labeled training data is often a significant challenge.

What it is not

AI has shortcomings that one must consider before employment. Neither symbolic nor non-symbolic AI approaches cope well with novel situations and require a human to re-engineer (symbolic) or retrain (non-symbolic) the algorithms. Symbolic, knowledge-engineered systems may contain underlying biases of the individual(s) who encode the system. Training data sets for non-symbolic approaches may contain biases that are not representative of the operational environment. These biases lead to either false positives, or worse, false negatives when the system is deployed. Such situations ultimately erode a user's trust, especially if the user has no avenue to investigate how the underlying AI arrived at its decision. This problem can be exasperated with non-symbolic approaches as they are steeped in mathematical equations. The underlying reasoning that supports inferences is inherently uninterpretable. Users of these systems do not have a way to interact with the system, question it, and receive an explanation as to how it arrived at its decision.

There are also cybersecurity concerns related to the employment of AI. Non-symbolic, ML systems can be spoofed by introducing imperceptible variations into the input, thereby causing a cybersecurity product to change its classification of a malicious document from "bad" to "good." Because both symbolic and non-symbolic AI systems are designed to make progress towards multiple goals, cyber-attackers could inject data into the environment that leads to goal conflicts, resulting in undesirable behaviors. For example, in swarm systems, modifying the perceived goal of a single agent could cause the entire multi-agent swarm to act unpredictably. To address these issues, AI systems will need to bring situational context to bear and use that context to determine whether the situation is in line with expectations. Outcomes that do not fit expectations would then cued for further investigation and provide opportunities for additional learning.

Where we are

We need synthetic agents that can act as our teammates in cyberspace, particularly in Defensive Cyberspace Operations (DCO). The task is daunting because of the breadth of capabilities that such an agent would need. Below are some of the most important ones.

- **Sense.** Though we have many ML systems that can sense a variety of phenomena in cyberspace, these platforms are narrowly focused on specific applications. What we need is a generalized ability to ingest and integrate information from multiple sources for a variety of purposes. Ideally, humans and synthetic agents would use the same tools for sensing the environment so that sensors can be operated by either.
- **Think.** Autonomous agents make decisions based on what they sense in their environment combined with what they already know. At a minimum, the agents must respond appropriately to events for which they have an "approved solution" and investigate ambiguous situations when the situational context does not meet their expectations to understand and make adjustments. They should also experiment with novel solutions to new situations, learning what works and what doesn't along the way.
- **Communicate.** If they are to be true teammates, our synthetic counterparts must know when and how to share information with their human counterparts. Their speed and capacity will preclude sharing everything in real time, but they must spontaneously reach out to their human supervisors when encountering specific situations and before embarking on risky exploratory behavior. The idea is to move the human to be ***on*** the loop instead of ***in*** it. Obviously, the agent must be able to respond to orders and questions from its human teammates and explain what it is doing and why in terms humans can understand.
- **Act.** It does us no good for agents to detect incidents and then not be able to respond autonomously. Clearly, we'd want to put bounds on risky responses, but faced with the eventuality of synthetic attackers, we can't afford to wait on significantly slower human responses. This act capability is the counterpart of the sense capability discussed earlier. Similar to sensing, the agents should influence their environments using tools that they could exchange with their human teammates at any point in an operation.
- **Learn.** In many ways, this is the most mature of the five requirements. We have a variety of learning mechanisms for both symbolic and non-symbolic AI that allow autonomous agents to improve their performance and adapt to changing environmental conditions. Still, we have some work to do improving agents interactions with a variety of human and synthetic teammates. We also need them to learn the adversary's behavior at a cognitive level rather than just recognizing their tools and left-behind indicators.

For the past two years, we have led research work on developing prototypes of offensive, defensive, and generic cyberspace agents that explore some of the building blocks required to provide these five capabilities. This family of synthetic teammates, called Cyber Cognitive (CyCog) agents is depicted in Figure 2. They all share a core system (CyCog) that they each refine with additional capabilities; this allows for time savings through software reuse.

The attacker version, CyCog-A, is intended for penetration testing and adversarial emulation during training events. CyCog-D is its defensive counterpart, which has only been used in support of training but already incorporates features that would allow it to effectively modify firewall and intrusion detection system (IDS) configurations in response to attacks. Finally, we are developing generic persona agents (CyCog-P) that behave as cyberspace denizens modeled after real users of a network under study.

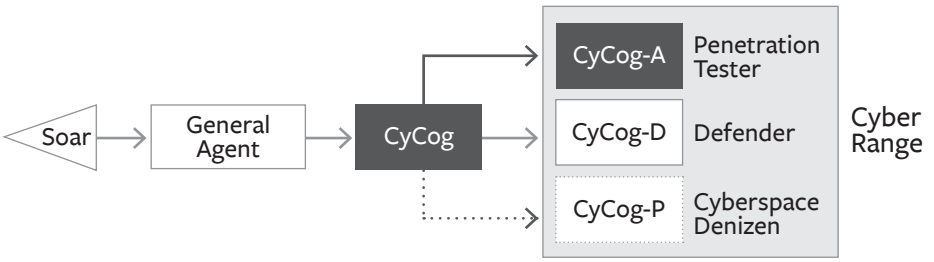

Figure 2: Genealogy of Cyber Cognitive (CyCog) agents

Because these agents are built on the Soar cognitive architecture, primarily a symbolic form of AI with some inherent non-symbolic features to support reinforcement learning and spatial reasoning, their cognitive models are inherently understandable by humans. This feature is illustrated in Figure 3, where we show an example goal tree (i.e., decision-making process) with a leaf node indicating an actual on-net action (i.e., sending a phishing email). This representation is easy to follow as the behavior model follows the cognitive processes of an attacker. Recall, however, that the bane of symbolic AI is this need for human-built models. Wouldn't it be possible to build AI systems that autonomously generate these?

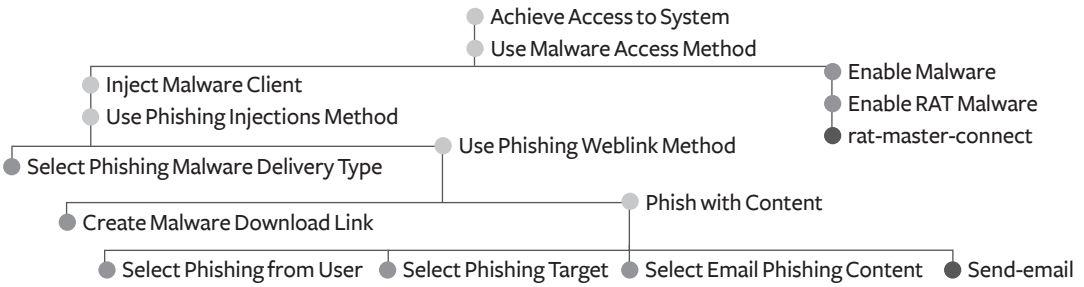

Figure 3: Partial CyCog-A goal tree showing a successful phishing attack

As an initial exploration of this possibility, we are in the early phases of a research project for the Office of Naval Research that seeks to develop ML modules that observe cyberspace activities, piece them together into procedures, and finds interesting (i.e., anomalous) ones. Codenamed *Twiner,* this system will allow us to reason over the three layers of cyberspace as defined by the U.S. DoD: persona, logical, and physical. By doing so, we believe we'll be able to detect behavioral patterns that would not be evident by looking at just one of these layers. This project will lay the groundwork for the autonomous identification of adversarial procedures and techniques, which, in turn, will allow us to automatically generate behavioral models and thus overcome one of the great limitations of symbolic systems.

The road ahead

Followed to its logical conclusion, Twiner and CyCog exemplify the symbiosis that results from leveraging both symbolic and non-symbolic approaches. Each plays to its own strengths while mitigating the limitations of the other. We already discussed how we could build non-symbolic AI systems that could observe cyberspace activities and build behavioral models for the symbolic AI agents. Conversely, these agents would be able to reason and act over a much broader set of observations, problems, and solutions than a non-symbolic AI system ever could.

A key takeaway from this paper is that to realize the full potential of AI, we must integrate its various forms in order to offset the limitations of each. No one approach will be sufficient because each approach is optimized for one specific set of problems at the expense of others. We can see this sort of integration in our own brains. According to Daniel Kahneman in his bestselling book *Thinking, Fast and Slow*, our brains leverage two systems: system 1 is fast, automatic and very task-specific (analogous to non-symbolic AI), and system 2 is slower, effortful and able to make complex decisions (analogous to symbolic AI). We all have cognitive mechanisms that allow us to switch from one to the other, and so should our synthetic teammates.

Good bedfellows

This paper has chronicled where we started, where we are, and where we should be going in the development of AI for cyberspace. Along the way, we have provided a fair amount of details about AI and ML. So, with all this in mind, how can you tell when someone is using these terms appropriately and when they are just hyping and overusing the terms? Below are three ideas you can try the next time someone wants to sell you on their version of AI.

Ask lots of questions. This may sound obvious, but many of us hesitate to ask questions when we think we know very little about a topic. We also tend to assume that if others speak authoritatively, then they must know what they're talking about. Even if you do not fully understand the responses (and you should keep drilling until you find something that makes sense), the manner in which others respond to your probing questions will tell you a lot

about their level of knowledge and how their solution works. Keep in mind that they usually cannot tell how much you know about AI, so they may get uncomfortable and be betrayed by their speech and body language. For better results, combine questions with the next suggestion: term familiarity.

Be familiar with key terms. Recall that, at their core, non-symbolic (a.k.a. ML) techniques are most commonly used for three purposes: *classification* (e.g., *k-Nearest Neighbors* or KNN), *clustering* (e.g., *k-Means*), and *prediction* (e.g., *regression*). They all work on *features* of the data they analyze (e.g., source IP address, interarrival rate), typically require large *data sets*, and always have a non-zero *false positive* error rate. Conversely, symbolic techniques require modeling of human knowledge that typically involves *cognitive modeling* and/or *task analysis*. As a starting point, you can make a list of all the italicized terms in the preceding text and learn a bit more about them. Even a summary understanding of them will go a long way in helping you tell when someone is trying to bamboozle you.

Call a friend. Most of us cultivate a diverse professional social network. Odds are that you know a couple of people who know enough about AI to help you separate the wheat from the chaff. (If you do not, this would be a perfect time to start making such friends.) Find them and ask for their opinion. Better yet, bring them along when you meet whoever will present to you their AI-powered solution. If the presenter lacks honesty or expertise, your friend should be able to tell right away even if you can't. Otherwise, it will be helpful to have someone who can help you translate the lingo, so you understand what is happening under the hood.

NOTES

1. As with all taxonomy classifications, such as the one in Figure 1, variations exist. For example, a symbolic, rule-based system can have non-symbolic mechanisms (e.g. reinforcement learning) and a non-symbolic approach can use symbols such as a neural network that outputs a classification label such as 'cat', 'dog' from an ingested image of pixels.

2. *Offline* learning occurs in an environment separate from where the system is deployed. *Online* learning is when the system learns as it is operating in its intended environment.

3. Despite using the same letter for their namesake variable, KNN and k-Means are entirely different algorithms for different purposes and with different requirements. The details, however, are beyond the scope of this paper.

4. It is not clear how many layers in a neural network one has to have before it is considered a deep neural network. Ten layers or more is often considered the benchmark.