INSTITUTE FOR DEFENSE ANALYSES

Utility of Artificial Intelligence and Machine Learning in Cybersecurity

Francisco L. Loaiza, *Project Leader*

John D. Birdwell

George L. Kennedy

Dale Visser

June 2019

IDA Non-Standard
NS D-10694

INSTITUTE FOR DEFENSE ANALYSES
4850 Mark Center Drive
Alexandria, Virginia 22311-1882

The Institute for Defense Analyses is a non-profit corporation that operates three federally funded research and development centers to provide objective analyses of national security issues, particularly those requiring scientific and technical expertise, and conduct related research on other national challenges.

About This Publication

This work was conducted by the Institute for Defense Analyses (IDA) under contract HQ0034-14-D-0001, Task DI-5-4630, "Army Software Marketplace Acquisition Strategy," for Program Executive Office Enterprise Information Systems (PEO EIS). The views, opinions, and findings should not be construed as representing the official position of either the Department of Defense or the sponsoring organization.

Acknowledgments

David A. Wheeler

For more information:

Francisco L. Loaiza, Project Leader
floaiza@ida.org, 703-845-6876

Margaret E. Myers, Director, Information Technology and Systems Division
mmyers@ida.org, 703-578-2782

INSTITUTE FOR DEFENSE ANALYSES

IDA Non-Standard NS D-10694

Utility of Artificial Intelligence and Machine Learning in Cybersecurity

Francisco L. Loaiza, *Project Leader*

John D. Birdwell

George L. Kennedy

Dale Visser

1. Executive Summary

This work was performed in support of the Army Software Marketplace Acquisition Strategy project (DI-5-4630). The Army has "identified the need to reduce costs and delivery time across the enterprise related to software generation, access, management, and sustainment. The envisioned solution calls for the deployment of a centralized software marketplace, coupled with the development of processes, procedures and governance for the submission and approval of new application software, and overall management of the software repository."

This work and paper partly fulfill the following paragraphs of the statement of work in the project description: (3g), which states the intent to "evaluate technical options and alternatives … for standing up an enterprise-level Army Application Development Environment (ADE) that supports development for the full range of software platforms…"; (3j), which states the intent to "investigate options for automating the application vetting process using commercial workflow tools and software testing best practices"; and deliverable (4d), "a draft report on maturity and applicability of options that can support the creation of an Army ADE."

The paper specifically discusses selected publications that relate artificial intelligence (AI) in general, or machine learning (ML) in particular, to cybersecurity and specifically to the cybersecurity of system development and life cycle environments (SDLE)[1] and their products. Some of the papers covered in this document are surveys that provide an overview of the area, and the present report is a meta-analysis that relies significantly upon surveys of the available literature. This approach was necessary because of the very large volume of publications in this field of research. A few papers are cautionary, pointing out that systems trained via AI or ML can be fooled; one of these papers investigates methods for designing classifiers that exhibit resilience to adversarial actions and points to other literature in this emerging field. Several references provide guidance for those who might

[1] The term "system development and life cycle environment" covers the infrastructure, processes and procedures, tools, and personnel involved in the design, development, and maintenance of a system, from conceptualization or initiation to removal from final service and disposal. *NIST Special Publication 800-64* discusses the various phases of this life cycle (see Kissel, Richard, Kevin Stine, Matthew Scholl, Hart Rossman, Jim Fahlsing, and Jessica Gulick 2008). The acronym "SDLE" is often used to stand for "software development and life cycle environment," and although this report primarily deals with issues relating to software development, "system development…" is used herein to emphasize the breadth of issues related to life cycle management and to conform to the NIST publication.

be interested in further reading across the breadth of the field. A few relevant publications from the NIST 800 series are included to provide an appropriate setting for the discussion of AI/ML as applied to cybersecurity.

The publications reviewed in this paper were found and selected using a variety of sources. Highly cited survey articles were found using Google Scholar,[2] and the citations in those articles led to other works via library searches. Other articles were found using combinations of search terms involving cybersecurity, AI, and ML. Patents were located and retrieved using Google's patent search capability.[3] Relevant books were identified using searches of library and other databases. There is an immense body of published work relating to AI and ML that extends over six decades (and further back, if one considers publications by Turing,[4] von Neumann,[5] and various other researchers who had combined backgrounds in physics, mathematics, engineering, and the biological sciences[6]). Although the corpus of literature relating to cybersecurity has a shorter time line (about three decades), there can be no claim that a brief review such as this one can be exhaustive. The authors hope that their apology for leaving out the reader's favorite references will be accepted, and that the discussions and observations that are offered can assist in determinations and selections of technologies, products, and methods for incorporation in high-quality SDLEs.

Cybersecurity is a very broad term referring to essentially everything that has a bearing upon protection of cyber resources. The term *artificial intelligence* dates from a 1955 proposal[7] for a summer research workshop held at Dartmouth in 1956 by McCarthy, Minsky, Rochester, and Shannon, and there is no generally accepted definition of what constitutes AI. ML is a subset of AI, but again, the definition is nebulous. These three fields have ill-defined boundaries.

For the purposes of this report, we define AI as heuristic methods (as opposed to algorithms firmly grounded in mathematics or fields such as mathematical optimization or nonlinear programming) designed to allow a computational system to function in a manner

[2] https://scholar.google.com/.

[3] https://www.google.com/?tbm=pts

[4] See (Turing, A. M. 2009), a reprint of Turing's article (*Mind*, 59:433-460 1950).

[5] See the discussion in (Mühlenbein, Heinz 2009).

[6] See, for example, McCulloch, Warren S, and Walter Pitts (1990), reprinted from the *Bulletin of Mathematical Biophysics*, Vol. 5, pp. 115-133 (1943).

[7] See http://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html for the text of the proposal. See also McCarthy, John, Marvin L. Minsky, Nathaniel Rochester, and Claude E. Shannon, "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence. August 31, 1955." 2006. AI Magazine 27 (4).

similar to a skilled human practitioner.[8,9] ML technologies utilize examples of the items one wants to categorize—whether labeled (for supervised learning), unlabeled (for unsupervised learning), or mixed—to train a computational system to behave in a desired manner. As stated earlier, ML can be viewed as a subset of AI; indeed, the definition of AI has continued to bloat over the decades since the 1950s to include (or at least substantially overlap with) fields that used to be considered distinct, such as heuristic optimization (simulated annealing, genetic algorithms, and genetic programming), pattern recognition, pattern classification, some aspects of signal processing, and image processing. Without arguing the merits of this point of view, this report merely considers and includes technologies that appear both useful and are related to AI or ML.

This report is a partial response to the question, "Do artificial intelligence and machine learning technologies provide opportunities to improve the cybersecurity of SDLEs and their products?" This question was motivated in part by the perception that automation plays a very significant role in the "dark side" of cybersecurity: the tools used to exploit information systems and organizations and to compromise their functions and exfiltrate their information. The time delay from a successful exploit to utilization of the compromised system or information can be extraordinarily short. (Tucker 2019; "CrowdStrike 2018 Global Threat Report" n.d.) It has become imperative that organizations that manage SDLEs achieve correspondingly short threat response time delays.

A. Summary of Findings

- AI/ML is viewed as a necessary response to the continuing growth in the number and complexity of threats, the evolving nature of threats, and the need for rapid (and therefore substantially automatic) responses to detected threats.
- It is clear from the literature that an expansive and broad definition of AI and ML can and should be applied in this field, encompassing a variety of methods that have developed over many decades (and in one case – naïve Bayes – centuries), have demonstrated effectiveness, and are currently in use.
- The primary targets for AI and ML application at present are intrusion detection (network-based attacks), phishing and spam (emails), threat detection and

[8] Given the coverage of "deep learning" in both technical forums and the popular press, one might assume that AI and ML are deep learning. Deep learning has earned its reputation from its recent successes in the fields of facial recognition, speech recognition, machine translation, and autonomous vehicles, but AI and ML are much broader fields. See Appendix A for a discussion of a 2015 survey paper on deep learning (LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton 2015) to gain a better understanding of deep learning.

[9] A JASON report loosely defines AI as "the ability of machines (computers) to perform tasks that humans do with their brains". ("Perspectives on Research in Artificial Intelligence and Artificial General Intelligence Relevant to DoD" 2017)

characterization (malicious code, largely because of their ability to deal with large number of variants of each threat), and user behavioral modeling. A rapidly emerging target for application is automated vulnerability testing and intrusion defense.

- Intrusion detection systems typically utilize hybrid approaches that combine several methods: signature-based methods for rapid detection of known threats with low false alarm rates (FARs) and anomaly-based methods to model normal behaviors and to flag deviations from the model's expectations.
- It appears that essentially all the vendors of cybersecurity products are working to adopt AI/ML components in their products.
- The paucity of datasets for research and development (R&D) in this area is a problem. Some vendors have large volumes of reasonably current data available (for example, network switch/infrastructure vendors, and providers of anti-virus and network/computer monitor software and systems). However, widely available datasets are extremely dated (DARPA 1998 and 1999, and KDD 1999 data), and the characteristics and volume of attacks have significantly changed since that time.
- There are indications that AI- and ML-based approaches may be easily spoofed or bypassed. Several published examples are not directly related to cybersecurity, but instead to spoofing systems that process image data or audio data, which appears to be easy. Some examples, however, are in the cybersecurity field.

B. Conclusions

Little development or customization effort is likely to be needed to apply AI and ML in the more mature application areas such as intrusion detection, activity modeling and abnormality detection, and next generation anti-virus technologies. The DoD will probably dictate what all systems must utilize to protect and monitor resources and users, and there probably will not be much flexibility. The majority of these capabilities will probably be deployed using commercial systems, possibly developed for government use and supported by large vendors. One reason for this is the rapid pace of evolution of threats. Keeping up with adversaries requires large and active organizations and consortia. The key will be to ensure that the essential bases are covered: firewall protection, including stateful inspection and control of network traffic across defined boundaries; intrusion detection and response; network, server, and endpoint monitoring, including user activity monitoring; and anti-virus and other malware protection.

The potentially overwhelming volume of measurement data that cybersecurity tools generate and the speed at which this volume grows create risks for any organization that

implements these protective systems. AI and ML offer some opportunities to mitigate these risks. Examples include intrusion detection systems that classify network traffic and measurements obtained from equipment connected to the network, systems that attempt to identify variants of known malware, and user activity monitoring systems that alert to anomalous behaviors. One important way in which emerging technologies such as AI and ML should be useful is in cutting through the volume of data and finding indicators of compromise using correlations across data sources. These systems would assist human analysts by elevating or alerting them to significant events that require responses without overwhelming the organization with false alarms or other spurious indicators. However, systems that incorporate advanced algorithms such as AI and ML must be properly designed to accept and process the high arrival rates of network and measurement data, as must the software-based sensors that collect data from hosts attached to the network, to ensure that the performance of both network segments and hosts is not unduly compromised.

Emerging technologies, including AI/ML, should be adopted to test systems (software, hardware, or both). AI and ML would be useful for automating testing for vulnerabilities, automating patching, and helping to enforce product quality standards. The emerging technologies will need to be carefully integrated with other systems that support and enforce the SDLE, including configuration management, test management, bug tracking, and workflow management tools. Unlike the developed application areas, significant resources will probably be needed to develop and integrate tools, some of which are likely to be open source products, into an effective SDLE. There are already some emerging commercial entities and, with time, commercial products will likely mature in these application areas.

C. Structure of Report

Chapter 2 of this report provides a summary of information from recently published cybersecurity assessments, detailing where things currently stand. These assessments are typically provided annually by several of the major commercial vendors.

Chapter 3 provides short discussions of several publications that relate AI or ML to cybersecurity in the relatively more developed contexts of intrusion detection systems (both signature-based and anomaly-based) and insider threats. The section begins with selected general-interest articles, continues with discussions of several review articles, and concludes with articles that address adversarial adoption of AI and ML technologies.

Chapter 4 provides a review of the Defense Advanced Research Projects Agency's (DARPA's) Cyber Grand Challenge (a "capture the flag" (CTF) tournament), which took place from 2014 to 2016, and the resulting follow-on efforts. This section provides support for the thesis that the emerging automated testing and repair capabilities should be incorporated into a well-designed SDLE.

Chapter 5 provides brief overviews of several books covering AI and ML in the context of cybersecurity and is followed by a brief summary of this report in Chapter 6.

Chapter 7 lists references, and Appendix A discusses a recent survey paper on deep learning.

2. Information from Recently Published Cybersecurity Assessments

Several large vendors produce annual assessments related to cybersecurity, and it is useful to review their findings.

Cisco ("Cisco 2018 Annual Cybersecurity Report" 2018) has provided a concise summary of the current state of affairs, estimating that 53% of attacks cause damages of $500,000 or more: "We know that attackers are evolving and adapting their techniques at a faster pace than defenders. They are also weaponizing and field testing their exploits, evasion strategies, and skills so they can launch attacks of increasing magnitude. When adversaries inevitably strike their targets, will defenders in the impacted organizations be prepared, and how quickly can they recover? That depends largely on the steps they're taking today to strengthen their security posture" (p. 46). An observation in the Cisco report is relevant to mobile device security: "The most challenging areas and functions to defend are mobile devices, data in the public cloud, and user behavior" (p. 47). Though one would appreciate finding a single vendor or product that can protect the environment, the reality according to Cisco is that in "2017, 25 percent of security professionals said they used products from 11 to 20 vendors" (p. 48), with "16 percent" using "anywhere from 21 to 50 vendors" (p. 48), and that "[s]ecurity teams face challenges in orchestrating multiple vendor alerts." (p. 49).

HP Development Company, LP emphasized the importance of AI and ML in their 2018 Cybersecurity Guide ("Hackers and Defenders Harness Design and Machine Learning" 2018). One point made in this report is that "[f]or every 100 lines of source code written … there's typically one defect … All a hacker has to do is find a defect that gives them entry to the system" (p. 6). The final section of the report is titled "Through the Looking Glass: Machine Learning and Artificial Intelligence," in which the authors state, "The days of malware scanning as the main tool against attackers are limited. … On the defenders' side, data plus machine learning are starting to be leveraged for automated network analysis, which crunches data from the constant stream flowing into, out of, and across the business network, looking for anomalies. Such computing capabilities will one day watch for spikes in processor activity, for instance, which could signal that a problem is starting. On the attackers' side, of course, machine learning can be used to automate network probing, scanning and scrubbing." (p. 19) According to a person quoted in the report, "We're going to live in a world of AI-enabled smart attacks." (p. 20).

The 2019 IBM X-Force Threat Intelligence report ("IBM X-Force Threat Intelligence Index 2019" 2019) emphasized phishing attacks, human errors and vulnerabilities, and threat vectors aimed at web-based email and cloud services. Automation using AI, ML, and more traditional methods can significantly reduce risks associated with these threats

through security education to standardized processes, automated tracking and reporting, and ML and natural language processing (NLP) for identification of phishing attempts.

The IBM report also discussed the Spectre and Meltdown attacks, which targeted Intel, and later AMD, hardware and firmware vulnerabilities associated with "speculative execution." (p. 5 and 27) Exploitation of hardware vulnerabilities is not new, but new threats in this class will continue to emerge. What has changed is the rapidity with which exploits are developed, deployed, and refined (Abu-Ghazaleh, Ponomarev, and Evtyushkin 2019). It is extremely difficult to respond to this class of threats; for example, a comprehensive solution to Spectre and Meltdown requires replacement of computer equipment, and discovery of the existence of these threats requires detailed research by highly skilled security researchers.

Oracle and KPMG issued a joint 2019 report that covers several themes, including the complexities cloud-based resources introduce to cybersecurity, the problems associated with patching systems to keep up with emerging threats and the potential of automation, multifactor authentication, and utilization of machine learning-powered analytics. ("Oracle and KPMG Cloud Threat Report 2019" 2019) With regard to ML, the report states the following:

> Advances in artificial intelligence, specifically machine learning, have had highly promising results in improving the efficacy of cybersecurity technologies such as endpoint security to detect and prevent new and previously unseen-in-the-wild malware. Machine learning is now incorporated into seemingly every new cybersecurity control intended to protect core-to-edge applications and data assets from compromise. In addition, some companies that have a requirement to train machine learning algorithms on industry-specific data sets, such as sensor data from smart automobiles, employ their own data scientist. These organic and integrated use cases have appreciably increased the use of machine learning for cybersecurity purposes over the last year. In fact, more than half of the respondents report they are using machine learning technology for cybersecurity purposes to some degree, up from 47% in 2018. North American companies are ahead of the curve with more intense usage of machine learning-based controls, per the 29% of those companies leveraging machine learning extensively. This level of adoption has made machine learning a foundational cybersecurity technology and especially applicable for certain use cases (p. 53).

The respondents to a survey conducted by Oracle and KPMG indicated that ML will be primarily used in security analytics and operations, as part of identity and access management, and user behavior analytics, and as part of a cloud security strategy. The primary benefits the respondents anticipated were improvements to investigation of security alerts, improved accuracy, reduced false positive rates, the elimination of more compute-intensive detection techniques, detection of zero-day threats, and the ability to

better utilize junior analysts. (These benefits are perhaps more of a wish list than realistic expectations; it is important to maintain a healthy level of skepticism.)

The following bullets summarize findings in these reports that have been discussed above:

- "…attackers are evolving and adapting their techniques at a faster pace than defenders" (Cisco, p. 46).
- "The most challenging areas and functions to defend are mobile devices, data in the public cloud, and user behavior" (Cisco, p. 47).
- "Security teams face challenges in orchestrating multiple vendor alerts" (Cisco, p. 49).
- "[D]ata plus machine learning are starting to be leveraged for automated network analysis" by defenders (HP, p. 19).
- "We're going to live in a world of AI-enabled smart attacks" (HP, p. 20).
- One emphasis is upon phishing attacks, human errors and vulnerabilities, and threat vectors aimed ad web-based email and cloud services (IBM).
- Exploitation of hardware vulnerabilities, as seen in the Spectre and Meltdown attacks, is not new, but new threats in this class will continue to emerge. What has changed is the rapidity with which exploits are developed, deployed – and refined (IBM).
- Cloud-based resources introduce additional complexity in cybersecurity (Oracle/KPMG).
- Timely patching and path automation are needed to combat the growing complexity of the threat landscape (Oracle/KPMG).
- "Machine learning is now incorporated into seemingly every new cybersecurity control intended to protect core-to-edge applications and data assets from compromise" (Oracle/KPMG, p. 53).
- "…more than half of the respondents [to a survey] report they are using machine learning technology for cybersecurity purposes to some degree, up from 47% in 2018" (Oracle/KPMG, p. 53).
- A survey indicated that the primary uses of ML will be in security analytics and operations, as part of identity and access management, and user behavior analytics, and as part of a cloud security strategy (Oracle/KPMG).

3. Discussion of Referenced Papers

The following discussions summarize the findings of publications that have been reviewed for this report. The publications are grouped into three broad topics:

- General-interest news articles
- Survey articles
- Cybersecurity risks associated with AI and ML and adversarial machine learning

Chapter 5 of this report provides very brief summaries of several books and reports that were discovered as the review of the literature relating AI and ML to cybersecurity progressed.

The general-interest news articles provide introductory discussions of topics relevant to the field and should be considered "light reading". This report's primary focus is a meta-analysis of the field based upon a survey of review articles. The reliance upon review articles was necessitated by the very large quantity of articles in the technical literature in this field. The large bibliographic listings published in the review papers provide evidence of this explosion of literature.

As the survey of the literature was conducted, articles were discovered that point out risks associated with the emerging utilization of AI and ML. Most of these articles are not specific to cybersecurity, but the generalization of the risks to cybersecurity applications is reasonably obvious.

The articles and books within each group are discussed in alphabetical order according to their bibliographic entries. A complete bibliography of the reviewed articles, together with other articles mentioned in this report, follows.

A. General-Interest News Articles

These are short articles from the popular and industry press that provide very brief overviews of topics relating AI and ML to cybersecurity.

Ciccatelli, Amanda. 2016. "Will Artificial Intelligence Revolutionize Cybersecurity?" *Inside Counsel Breaking News*, June 27, 2016. General OneFile.

This is a news article that is light on details. However, it does mention an analogy of nature-inspired AI to a biological immune system that might detect and inoculate against novel threats, the potential of evolutionary computation in ML, and the possibility of "multi-agent AI techniques."

Greengard, Samuel. 2016. "Cybersecurity Gets Smart." *Communications of the ACM, 59* (5): 29–31.

This is a general interest magazine article rather than a technical research article. The thesis is that the "traditional approach of using signature-based malware detection, heuristics, and tools such as firewalls and data loss prevention (DLP) simply is not getting the job done. … Traditional security methods aren't keeping up with cyberthieves" (p. 29). AI methods such as "big data, pattern mapping and matching, cognitive computing, and deep learning methods that simulate the way the human mind works" (p. 29) are being explored by researchers as ways to defend information resources. "The goal … is to better identify suspicious patterns and behavior" (p. 29). "Manual approaches and signature-based approaches are no longer effective" because of the large and increasing number of threats (p. 29). Problems include the "growing prevalence of zero-day attacks …, polymorphous malware …, viruses, Trojan horses … and graphics processing units" (p. 29). In addition, "firewalls have become less effective as cloud computing and APIs string together data across enterprise boundaries" (p. 29). "[S]ecurity threats ranging from social engineering … to botnet .. [are] more difficult to pinpoint and block because they use cloaking techniques and alias IP addresses" (p. 30).

Techniques being explored include:

> [c]ognitive computing … using … natural language processing to analyze code and data on a continuous basis. As a result, it is better able to build, maintain, and update algorithms that better detect cyberattacks, including Advanced Persistent Threats (APTs) that rely on long, slow, continuous probing at an almost-imperceptible level in order to carry out a cyberattack. (p. 30)

"One company at the vanguard of AI is Tel Aviv, Israel-based Deep Instinct, which has introduced security software that uses an artificial neural network (ANN) to digest huge volumes of data and put it to use quickly and effectively" (p. 30). According to the Chief Technology Officer of Deep Instinct, a large fraction of new malware is very similar to previously detected malware (i.e., there is less than 2% difference in code), but that is sufficient to "throw off most conventional malware detection tools" (p. 30). Deep Instinct utilizes deep learning and claims a 98.8% detection rate compared to a 79% detection rate for the next best solution.[10] (Note that this is not an independent assessment.)

Another approach developed at Georgia Tech and licensed to Symantec uses "an algorithm that analyzes relationships with peer files using locality-sensitive hashing and graph mining, which clusters risks by probability" (p. 31). "Tests show the approach

[10] Note that as impressive as these percentages may be, when dealing with large numbers of instances, there is still a substantive residual subset that escapes detection. This could lead to a false sense of security, as one of those undetected instances of malware may be sufficient to cause severe damage to the organization.

identifies 99% of benign files and 79% of malicious files a week earlier than other technologies" (p. 31).

Garlan at Carnegie Mellon University studies "how to move beyond using AI merely on the detection side and harness[ing] it on the repair side. … Garlan says … [m]any of the answers already exist, … it is simply a matter of combining data points, crunching huge volumes of data, and rethinking interfaces to introduce more streamlined and functional machine-human interaction" (p. 31).

Hutson, Matthew. 2018. "Hackers Easily Fool Artificial Intelligences." *Science, 361* (6399): 215.

This is a general interest magazine article rather than a technical research article. AI systems are vulnerable to spoofing, where slight changes in inputs are introduced to produce dramatically different outputs (for example, classifications).[11] Examples include a modified image of a stop sign that was identified as a speed limit sign and a modified audio recording of a voice saying "without the data set the article is useless" that was identified as "OK Google, browse the evil.com." (p. 215) Demonstrations have been performed with and without knowledge of the implementation of the AI system. For example, it is possible (though more difficult) to perform a black-box attack without *a priori* knowledge of a system's implementation details.

The implications for cybersecurity are that a threat could be masked to appear innocuous to an intrusion detection system or that classified or sensitive information transmitted out of a secure environment could be masked. Steganography is an example of this (transmission of sensitive information within an image or other document), but analysis may be more challenging with an AI system, because the system's logic is not traditionally programmed or acquired.

Wilkins, Jonathan. 2018. "Is Artificial Intelligence a Help or Hindrance?" *Network Security*, 2018 (5): 18–19.

This is a news article written from the point of view of the industrial automation industry and the Internet of Things (IoT). The article argues that AI is beginning to be

[11] One can argue that spoofing is related to bias, which may be caused by an insufficiently robust training data set. However, it is always possible to introduce inputs not represented in a training set, and spoofing appears to point to a more fundamental issue. A neural network's parameter space typically has a very high dimension, and the neural network functions as an interpolator for inputs not previously observed. Just as in the case of interpolation of functions using orthogonal terms of a series, unexpected behaviors can occur between training data points, and these behaviors can grow more erratic as the dimension of the parameter space increases. One example is the "ringing" behavior that occurs near a function's discontinuities when the function is approximated using a Fourier series. It is reasonable to expect spoofing to always be possible regardless of the size of the training data set. A JASON report on artificial intelligence also discusses this issue. ("Perspectives on Research in Artificial Intelligence and Artificial General Intelligence Relevant to DoD." 2017)

incorporated into the toolkits used by cyber attackers, which could make cyber-attacks more powerful and efficient. The article states that "[i]n a survey taken during the Black Hat USA 2017 cyber-security conference, 62% of attendees predicted that the first AI-enhanced cyber-attack will happen in the next 12 months" (p. 18). The author advocates the use of AI in cybersecurity—whether or not incorporation of AI in defense strategies is the best solution, it is clear that it is quickly becoming part of the threat landscape. The author provides the example of Darktrace, a security company that "uses machine learning to create unique patterns of encryption for each machine and detect any abnormalities" (p. 19).

B. Survey Articles

These articles provide surveys of the literature and recent research results that relate AI or ML to cybersecurity.

Amit, Idan, John Matherly, William Hewlett, Zhi Xu, Yinnon Meshi, and Yigal Weinberger. 2018. "Machine Learning in Cyber-Security - Problems, Challenges and Data Sets."

This paper is not well written, but there is some value in the sections that identify ML challenges in cybersecurity and that discuss several data sets. Challenges include the perceived lack of labeled examples that can be utilized to train ML systems and the imbalance in the number of examples of malware to non-malware data—the latter typically being a much larger fraction of collected data. The latter issue imposes a large penalty upon false alarms produced by detection systems.

The authors point out that the data set issues are caused by companies' unwillingness to share collected data, which may also be due to privacy, contractual, damage to reputation, and legal constraints. Other issues include malware polymorphisms (derivative versions of an original malware or malware tool) and ambiguities and similarities among sites (IP addresses, or hosts) with which malware communicate.

Several bibliographic references are provided, but most are only briefly mentioned in the body of the paper.

Buczak, A. L., and E. Guven. 2016. "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection." *IEEE Communications Surveys Tutorials*, 18 (2): 1153–76.

This survey paper is an excellent reference that provides a good taxonomy of data mining and ML methods used across the research community for intrusion detection systems (IDS). The authors discuss data mining and ML, but it would be more accurate to

describe the surveyed literature as applications of ML using acquired data. The primary issues are as follows:

- the speed at which threats evolve, requiring either continuously adaptive IDS or frequent re-training;
- the large quantity of data relevant to IDS that can be acquired and its rapid arrival or acquisition rate; and
- the trade-offs between the level of detail in acquired data and the acquisition rate and between detail and privacy/security concerns.

In short, data mining to extract relevant features from acquired data is necessary, but this is normally an integral part of ML.

The authors define cybersecurity as "the set of technologies and processes designed to protect computers, networks, programs, and data from attack, unauthorized access, change, or destruction" (p. 1153). "Cyber security systems are composed of network security systems and computer (host) security systems. Each of these has, at a minimum, a firewall, antivirus software, and an intrusion detection system (IDS)" (p. 1153). The authors likely misspoke by saying "each"; the system must have, at a minimum, a firewall, antivirus software, and an IDS, any of which may comprise a network appliance or software or a component of a computer or host. A collection of components typically operates in a coordinated manner to implement these functions.

The authors distinguish between three types of cyber analytics that support IDS: misuse-based (or signature-based), anomaly-based, and hybrid. Misuse-based analytics are designed to detect known attacks without having a large rate of false positive alarms. They are not capable of detecting zero-day (never before seen) attacks. Anomaly-based analytics create a model of normal behavioral patterns and attempt to detect deviations from these patterns. They have the potential to detect novel attacks and generate signatures that can be utilized to detect similar future attacks. Hybrid analytics combine these two approaches. The authors report that the majority of the reviewed approaches were hybrids.

The paper discusses the two primary sources of data that can be used by an IDS: packet-level data, and NetFlow data. The difference between these data sets is granularity: The packet-level data set records information for each monitored network packet, whereas the NetFlow data records information for each transmitted or received data stream (possibly comprising many packets). Both data sources can be augmented by data collected from computers connected to a monitored network, including security logs and kernel (operating system) calls, and from network equipment (network logs). The paper makes strong arguments that an IDS should utilize these additional data sources, stating that "it is advantageous that an IDS be able to reach network- and kernel-level data. If only NetFlow (much easier to obtain and process) data are available for the IDS, these data must be

augmented by network-level data such as network sensors that generate additional features of packets or streams" (p. 1171).

A recurring issue in this and other reviewed publications is the paucity of reference data sources to support IDS research and development. Data sets published by DARPA (DARPA 1998 and DARPA 1999) and the data set utilized for the KDD Cup challenge in 1999 are the primary sources utilized by the majority of researchers. This is in part due to legal and privacy restrictions that limit the ability to collect and share network data, and in part due to commercial concerns (vendors that consider the data proprietary). However, even if data can be collected, significant resources (such as identifying and labeling packets or streams that are a consequence of malicious activities) are required before it can be useful to an IDS developer. As the paper states, "The fact that so many papers use the DARPA and KDD data sets is related to how difficult and time consuming it is to obtain a representative data set" (p. 1171).

The paper provides a discussion of standard metrics used to judge the quality of an IDS, including accuracy (proportion of data that are correctly classified), sensitivity (probability of attack detection), and FAR. In contrast to the paper by Xin et al. (2018), which appears to include much of the same information, this paper does not tabulate its surveyed IDS methods by accuracy, sensitivity, or FAR. It does, however, provide a better overview of the methods utilized by the various IDS.

The authors provide a reasonably comprehensive list of ML methods utilized in cybersecurity (primarily for IDS):

- ANNs
- Association rules and fuzzy association rules
- Bayesian networks
- Clustering
- Decision trees (DTs)
- Ensemble learning
- Evolutionary computation
- Hidden Markov models
- Inductive learning
- Naïve Bayes
- Sequential pattern matching
- Support vector machines (SVMs)

These methods have been developed over decades and applied in many fields of study since the 1950s (the Perceptron is an example and is the origin of ANNs), but in one case (naïve Bayes, which is the straightforward application of Bayes rule) dates from the 1700s. For each method, the authors present published applications in misuse detection, anomaly detection and hybrid detection, or all three together. The surveyed publications were selected by searches using Google Scholar for ("machine learning" AND "cyber") or ("data mining" AND "cyber"), with emphasis on highly cited papers, although "it was also recognized that this emphasis might overlook new and emerging techniques, so some of these papers were chosen also" (p. 1153).

An interesting outcome of this survey is that no method stood out as clearly the best approach for IDS. "Although some algorithms are accepted to be better performing than others, the performance of a particular ML algorithm is application and implementation dependent" (p. 1170). There was, however, a difference in the computational complexity across the methods, with clustering, nearest neighbor, and SVM methods having the lowest complexity. A critical issue is whether a method is "streaming capable"; that is, whether it can operate in real time as new data are received. Bayesian networks, clustering, naïve Bayes, and nearest neighbor methods were rated highly on this criterion, with ANN, association rules, hierarchical clustering, and sequence mining rated least capable of streaming.

The authors noted that although a high percentage of the surveyed papers present offline methods, realistic IDS must be online and capable of processing streaming data. Only four of the surveyed papers described their systems as online and operating in real time. Given the evolving characteristics of intrusion threats, it is essential that an IDS be adaptive and able to learn and respond to emerging threats. The authors state that "an online suitable method addresses, at a minimum, three factors: time complexity, incremental update capability, and generalization capacity. … A method should be close to roughly O(n log n) to be considered a streaming algorithm. … For the incremental update capability, the clustering algorithms, statistical methods … and ensemble models can easily be updated incrementally. … However, updates to ANNs, SVMs, or evolutionary models may cause complications. … A good generalization capability is required so that the trained model does not drastically deviate from the starting model when new input data are seen. Most of the state-of-the-art ML and DM methods have very good generalization ability" (p. 1170).

A final observation in the paper is the importance of access to reference data and the type of IDS that results: "…categorizing the studies with respect to the authors' affiliations reveals studies that built actual IDSs and employed real-world data captured from campus networks or Internet backbones. All of these studies appear to have used systems integrated with more than one ML method and several modules related to attack signature capture, signature database, etc." (p. 1171). In other words, an effective and functional IDS is most

likely not constructed using a single method or technique, but rather is a hybrid of multiple subsystems utilizing multiple data sources.

In their recommendations, the authors state the following:

> IDSs are usually hybrid and have anomaly detection and misuse-detection modules. The anomaly detection module classifies attack patterns with known signatures or extracts new signatures from the attack-labeled data comping from the anomaly module. Often, an anomaly detector is based on a clustering method. Among clustering algorithms, density-based methods … are the most versatile, easy to implement, less parameter or distribution dependent, and have high processing speeds. In anomaly detectors, one-class SVMs also perform well … . Among misuse detectors, because the signatures need to be captured, it is important that the classifier be able to generate readable signatures, such as branch features in a decision tree, genes in a genetic algorithm, rules in Association Rule Mining, or sequences in Sequence Mining. Therefore, black-box classifiers like ANNs and SVMs are not well suited for misuse detection. Several state-of-the-art ML and DM algorithms are suitable for misuse detection. Some of these methods are statistical such as Bayesian networks and HMMs; some are entropy-based such as decision trees; some are evolutionary such as genetic algorithms; some are ensemble methods like Random Forests; and some are based on association rules. … methods like Bayesian networks or HMMs may not be the strongest approach because the data do not have the properties that are the most appropriate for them. Evolutionary computation methods may take a long time to run and therefore may not be suitable for systems that train online. If the training data are scarce, Random Forests might have an advantage. If the attack capture is important, decision trees, evolutionary computation, and association rules can be useful. (p. 1173)

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. "Deep Learning." *Nature,* 521 (7553): 436–44.

This paper provides a survey and overview of deep learning research results and applications up to about 2015. A review of the paper is provided in Appendix A because of its value to a reader who needs a fairly detailed introduction to deep learning.

"Perspectives on Research in Artificial Intelligence and Artificial General Intelligence Relevant to DoD." 2017. JSR-16-Task-003. McLean, VA: The MITRE Corporation, JASON Program Office.

This report is the result of a JASON study sponsored by DoD/OSD/ASD(R&E). The study appears to have been triggered by the successes of multi-layer neural networks since the year 2000. The report states in its overview that this "phase-change re-energizing of a particular area of AI is the result of two evolutionary developments that together crossed a qualitative threshold: (i) fast hardware Graphics Processor Units (GPUs) allowing the

training of much larger—and especially deeper (i.e., more layers)—networks, and (ii) large labeled data sets (images, web queries, social networks, etc.) that could be used as training test-beds. This combination has given rise to the 'data-driven paradigm' of Deep Learning (DL) on deep neural networks (DNNs), especially with an architecture termed Convolutional Neural Networks (CNNs)" (p. 1). While acknowledging the successes attributed to deep learning, the authors are skeptical of its importance relative to other AI technologies:

> Deep Learning, based on DNNs trained on Big Data, is a tipping point in AI, evangelized by many fervent supporters. As a "dogma", DL has these beliefs: (i) Use of DNNs, often convolutional, at scale. (ii) Flat, numerical data representations. Inputs are vectors of reals. Internal data representations are tens to hundreds of millions of real-valued activations. (iii) Desirability of training on Big Data with few hard-wired model assumptions. DL seeks to learn everything from the data, believing that "data is where truth lies". (iv) The strong belief that an approximate answer is good enough. When a solution works, use it and don't ask too many questions about how it works.
>
> Nevertheless, the very real successes of the DL revolution may be overshadowing some other rapidly advancing areas in AI. The report discusses the successes of reinforcement learning (RL, which can be applied both to DL and other paradigms); graphical and Bayes models, especially with probabilistic programming languages; generative models that may allow training with much smaller data sets; and other kinds of probabilistic models such as those that have shown remarkable successes in question answering (e.g., IBM's Watson), machine translation, and robotics. While DL will certainly affect all of these fields, it is not the only or final answer. More likely, DL will become an essential building block in more complicated, hybrid AI architectures.
>
> …
>
> The so-called "ilities" are of particular importance to DoD applications: reliability, maintainability, accountability, verifiability, evolvability, attackability, and so forth. As a generalization, DL—in its current state of development—is weak on the "ilities". The full report discusses why, at a fundamental level, this is the case: DNNs are function approximators in very high dimensional spaces (e.g., millions of dimensions). The manifolds whose shape and extent they are attempting to approximate are almost unknowably intricate, leading to failure modes for which—currently—there is very little human intuition, and even less established engineering practice. (p. 2)

The report places deep learning methods within a historical context of AI and ML, which the report states "enjoys a special relationship with AI. It provides the foundational mathematical and statistical algorithms that are used in AI's application areas" (p. 5). The

report mentions specific ML algorithms, placing them in a "pre-modern" era, a "dawn of the modern" era, a "modern" era, and a "post-modern" era as follows (paraphrased from the report):

- Pre-modern era
  - Perceptrons
  - Expert systems
- Dawn of the modern era
  - Gaussian mixture models
  - k-mean clustering
  - Hidden Markov models (HMMs)
- Modern era
  - SVMs
  - Kernel methods
  - Ensemble methods such as "random forest"
  - Regularization methods based on Bayes priors
  - Hierarchical Bayes models
- Post-modern era
  - Deep neural networks (DNNs), including convolutional neural networks, when combined with big data (yielding so-called deep learning)
  - Graphical Bayes models, including statistical inference on large Bayes nets
  - Reinforcement learning (RL)

The report provides a good discussion of the technological components of deep learning, including its roots with the perceptron of the 1950s, the generic multi-layered neural network architecture, the use of nonlinearities such as the sigmoid function, training with back-propagation, and convolutional neural networks and pooling layers. Other detailed technical aspects of DNNs are discussed, including stochastic gradient descent, dropout methods, transfer learning, data augmentation, autoencoders, and recurrent neural networks using long- and short-term memory cells or gated recurrent unit cells, with references for additional information. The discussion of deep learning concludes with a summary of "the Big Data Deep Learning 'Dogma'":

> Use deep (where possible, very deep) neural nets. Use convolutional nets, even if you don't know why … . Adopt flat numerical data representations … Avoid the use of more complicated data structures. The model will

> discover any necessary structure in the data from its flat representation. Train with big (*really* big) data. … An approximate answer is usually good enough. When it works, it is not necessary to understand why or how. (p. 25)

Section 4 of the report discusses "Deep Learning and the 'Ilities'":

> … as an important caveat, the current cycle of progress in BD/DL has not systematically addressed the engineering 'ilities': reliability, maintainability, debug-ability, evolvability, fragility, attackability, and so forth. Further, it is not clear that the existing AI paradigm is immediately amenable to any sort of software engineering validation and verification. This is a serious issue, and is a potential roadblock to DoD's use of these modern AI systems, especially when considering the liability and accountability of using AI in lethal systems. (p. 27)

This section provides an analysis not only of these shortcomings of deep learning but also of why addressing these shortcomings may be difficult, along with a few examples to illustrate this point.

The remaining sections of the report discuss other promising areas of AI and the use of hardware acceleration for deep learning. The report concludes with a brief discussion of considerations specific to DoD and lists of findings and recommendations.

This report is well worth the effort necessary to read it. It can also serve as a counterpoint to the excellent review article on deep learning (LeCun, Yann, Yoshua Bengio, and Hinton 2015) discussed in Appendix A.

Taleqani, A. R., K. E. Nygard, R. Bridgelall, and J. Hough. 2018. "Machine Learning Approach to Cyber Security in Aviation."

The Taleqani paper discusses cyber risks to aviation operations and safety and the possibility that ML approaches can address these risks. The paper states that "… the aviation industry is highly susceptible to cyber-attacks. … According to the Directory of Strategy and Safety Management at the European Safety Agency, aviation systems were subject to an average of 1,000 attacks each month" (p. 147). Risks are associated with computer systems such as reservation systems, customer facing websites, and communications systems, including the aircraft communications addressing and reporting system (ACARS), next-generation air traffic control, and aircraft and air traffic control systems.

The paper divides air safety cybersecurity threats into two categories: phishing and network attacks; network attacks are further divided into eavesdropping, denial of service (DoS), man in the middle (MITM), and spoofing.

The paper referenced an interesting article (Whittaker, Ryner, and Nazif 2010) that "described the characteristics of a scalable classifier based on online gradient descent

logistic regression to detect phishing websites. It analyzed millions of pages a day to maintain Google's phishing blacklist automatically and finally generated on average a false positive rate below 0.1%" (p. 148). The approach described by Whittaker is implemented at scale; the article states: "During the first six months of 2009, our classifier evaluated hundreds of millions of pages, automatically blacklisting 165,382 phishing pages" (p. 2). At the time the paper was written, Google was using this technology to maintain the blacklist it published, which was utilized by the Firefox, Chrome, and Safari web browsers. The classifier is trained offline using a rolling window of data from the past three months. A proprietary ML system was used for training, but the authors compared the results to results obtained using a random forest approach. The authors stated their findings "suggest that both random forests and other online learning implementations would adequately substitute for our proprietary learning systems" (p. 6). The article references a patent that describes the ML method (Bem, Harik, and Levenberg 2007); note that there are a large number of subsequent patents, many assigned to Google, that cite this patent. Some of these provide enhanced methods for model generation using ML.

The Taleqani paper describes two other methods for detection and classification of phishing emails, but neither appears to perform as well as the method published by Whittaker. One uses a standard "Term Frequency-Inverse Document Frequency" (TF-IDF) method, while the other uses a combination of standard decision tree methods (inductive inference) and k-nearest neighbor (KNN) classification.

The Taleqani paper discusses various ML approaches for network-based attacks and references (Buczak and Guven 2016), which was discussed earlier in this paper.

Xin, Y., L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang. 2018. "Machine Learning and Deep Learning Methods for Cybersecurity." *IEEE Access,* 6: 35365–81.

This article is a survey of ML and deep learning for network analysis of intrusion detection. "[C]ommonly used network datasets" are also discussed. The paper categorizes attack types into three categories: misuse-based, anomaly-based, and hybrid detection. According to the authors, misuse-based attack detection relies upon attack signatures, does not generate excessive false alarms, requires frequent updates to signature databases, and cannot detect zero-day attacks. The authors say anomaly-based techniques have the capacity to detect zero-day attacks because they monitor for anomalous behaviors and can be used to profile normal activities and generate signatures for new attacks, but they have the potential for high FARs. The authors state that hybrid detection methods are combinations of the first two methods, are used to increase detection rates of known intrusions and reduce false positive rates, and that most ML/deep learning methods are hybrids.

This paper contains material that appears similar to an earlier survey paper (Buczak and Guven 2016), but more recent results have been added.

ML and deep learning algorithms discussed include SVMs, KNN, DT algorithms including those based upon Quinlan's ID3 and C4.5 and Breiman's CART, deep belief networks (DBN), recurrent neural networks (RNN), and convolutional neural networks (CNN). A selection of recent papers is discussed for each algorithm. Datasets discussed include the DARPA Intrusion Detection Data Sets, the KDD Cup 99 Dataset, the NSL-KDD Dataset, and the ADFA Dataset. All of these datasets except for the ADFA Dataset are based upon network traffic logs. The ADFA Dataset provides system call data for two operating systems, Windows and Linux. The list of references is reasonably extensive.

The results summarized in this survey article provide detection rates in the 80–100% range, with FARs from tenths of a percent to 10%. Of the 34 IDS methods that reported accuracy cited in this article, 28 reported accuracies of 90% or better, and 11 reported accuracies of 99% or better.

(Accuracy is defined as the percentage of classifications (positive or negative threats) that are correct. Note that if true positives—threats—are rare relative to benign items, accuracy can be misleading in that a highly accurate system can still have a large FAR.)

Only 16 studies reported precision (the percentage of alarms that were not false positives); 11 reported precisions of 90% or better, with 2 reporting 99% or better. The methods that utilized neural networks did not exhibit clear superiority (99% accuracy or precision, or better) to other methods; however, hybrid methods (using multiple detection technologies) appeared to have an advantage.

This paper did not point out the importance of having very high precision (or very low FARs) to the extent that it should. In a typical IDS installation on a network protected by one or more firewalls, malicious transactions are an extremely low percentage of total transactions. In this scenario, even quite small FARs (0.1% or less) may generate excessive false alarms. Only two of the 18 studies that reported FARs cited rates of 0.1% or less, and one of these is suspect (reporting 100% accuracy and 0% FAR). Three of the studies reported FARs near 10%—a level that is clearly unacceptable in a realistic setting. Of the three studies reporting the lowest FARs, two utilized DTs, and one utilized a convolutional neural network—not an impressive showing for neural network technologies.

C. Cybersecurity Risks Associated with AI and ML, and Adversarial Machine Learning

These articles address the potential for adversaries avoiding detection, causing misclassification, or posing other threats to systems that incorporate AI and ML by exploiting features of the AI/ML algorithms.

Brundage, Miles et al. 2018. "The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation."

This 100-page report is a result from a workshop held February 19-20, 2017, in Oxford, UK. Miles Brundage of the Future of Humanity Institute (FHI) and Shahar Avin of the Centre for the Study of Existential Risk (CSER) co-chaired the workshop. The focus of the workshop was the dual-use nature of AI and ML and on the implications this should have to the research and development community, the policy and political communities, and commercial and defense establishments. "The workshop was co-organized by FHI, CSER, and the Leverhulme Centre for the Future of Intelligence (CFI)." (Brundage, Miles et al. 2018, p. 75) Participation was broad, with attendance by individuals from OpenAI, University of Cambridge, University of Bath, Princeton University Center for Information Technology Policy, Yale Law School, Yale University, Electronic Frontier Foundation, University of Oxford, University of California at Berkeley, Microsoft Research, Google, DeepMind, Arizona State University, and University of Louisville, among others.

The report examines adversarial scenarios relating to digital security, physical security, and political security. The aspect of the work most relevant to the context of this review involves digital security. The report is written at a high level but provides references into the technical and policy literature. The primary take-away is that AI is a dual-use technology: "AI systems and the knowledge of how to design them can be put toward both civilian and military uses, and more broadly, toward beneficial and harmful ends" (p. 16). The report points out many issues, but one stands out: "Today's AI systems suffer from a number of novel unresolved vulnerabilities. These include data poisoning attacks (introducing training data that causes a learning system to make mistakes), adversarial examples (inputs designed to be misclassified by machine learning systems), and the exploitation of flaws in the design of autonomous systems' goals" (p. 17). The implications for the threat landscape are summarized: "…we expect attacks to typically be more effective, more finely targeted, more difficult to attribute, and more likely to exploit vulnerabilities in AI systems" (p. 18). In particular, "…if an actor begins to deploy novel AI systems, then they may open themselves up to attacks that specifically exploit these vulnerabilities. … we expect the attacks supported and enabled by progress in AI to be especially effective, finely targeted, difficult to attribute, and exploitative of vulnerabilities in AI systems" (pp. 20–21). One example discussed was spear phishing: "spear phishing is more effective than regular phishing, which does not involve tailoring messages to individuals, but it is relatively expensive and cannot be carried out en masse. More generic phishing attacks manage to be profitable despite very low success rates merely by virtue of their scale. By improving the frequency and scalability of certain attacks, including spear phishing, AI systems can render such trade-offs less acute. The upshot is that attackers can be expected to conduct more effective attacks with greater frequency and at a larger scale. The expected increase in the effectiveness of attacks also follows from the potential of AI

systems to exceed human capabilities" (p. 21). The authors expect increased automation and complexity of social engineering attacks; for example, "[a]s AI develops further, convincing chatbots may elicit human trust by engaging people in longer dialogues, and perhaps eventually masquerade visually as another person in a video chat" (p. 24).

The report also focuses upon the potential for AI and ML to introduce new vulnerabilities in systems they are designed to protect: "…we should expect attacks that exploit the vulnerabilities of AI systems to become more typical" (p. 22). The authors expect "[a]utomation of vulnerability discovery" and "[m]ore sophisticated automation of hacking" (p. 25). The availability of large datasets used to identify victims offers the opportunity to prioritize targets for cyber attacks using ML, and the AI used in applications, "especially [applications] in information security," may be exploited via data poisoning to "surreptitiously maim or create backdoors in consumer machine learning models," while black-box methods may be used to extract proprietary AI system capabilities, where "[t]he parameters of a remote AI system are inferred by systematically sending it inputs and observing its outputs" (p. 26).

The authors note that "[c]ybersecurity is an arena that will see early and enthusiastic deployment of AI technologies, both for offense and defense … AI is already being deployed for purposes such as anomaly and malware detection" (p. 31) The report states that important IT systems have often evolved into "sprawling behemoths, cobbled together from multiple different systems, under-maintained and – as a consequence – insecure. Because cybersecurity today is largely labor-constrained, it is ripe with opportunities for automation using AI. Increased use of AI for cyber defense, however, may introduce new risks… ." (p. 31). "To date, the publicly-disclosed use of AI for offensive purposes has been limited to experiments by 'white hat' researchers, who aim to increase security through finding vulnerabilities and suggesting solutions. However, the pace of progress in AI suggests the likelihood of cyber attacks leveraging machine learning capabilities in the wild soon, if they have not done so already" (p. 32). The report quotes Admiral Mike Rogers: "Artificial Intelligence and machine learning – I would argue – is foundational to the future of cybersecurity […] it is not the if, it's only the when to me" (p. 32).

The authors briefly mention the DARPA Cyber Grand Challenge contest of 2014–2016 (discussed in section 4 of this report) and state that "…the application of AI to the automation of software vulnerability discovery, while having positive applications … can likewise be used for malicious purposes to alleviate the labor constraints of attackers" (pp. 32–33). AI can be utilized to avoid detection, according to the authors, citing works that create "a machine learning model to automatically generate command and control domains that are indistinguishable from legitimate domains", use "reinforcement learning to create an intelligent agent capable of manipulating a malicious binary with the end goal of bypassing [next generation anti-virus] detection", and use "adversarial machine learning to craft malicious documents that could evade PDF malware classifiers" (p. 34). Large efforts

that utilize ML to identify threats and develop signatures for intrusion detection systems can also be used to generate malware; the authors cite "…services like Google's VirusTotal file analyzer [that] allows users to upload variants to a central site and be judged by 60+ different security tools. This feedback loop presents an opportunity to use AI to aid in crafting multiple variants of the same malicious code to determine which is most effective at evading security tools" (p. 34). The report states that "[w]hile the specific examples of AI applied to offensive cybersecurity … were developed by white hat researchers, we expect similar efforts by cybercriminals and state actors in the future as highly capable AI techniques become more widely distributed…" (p. 34).

Katzir, Ziv, and Yuval Elovici. 2018. "Quantifying the Resilience of Machine Learning Classifiers Used for Cyber Security." *Expert Systems with Applications* 92 (February): 419–29.

This paper models the susceptibility of a classifier to manipulation by an adversary as a mathematical game. An adversary incurs costs for manipulation, and that cost is balanced against the performance penalty a classifier incurs by utilizing less susceptible means of threat assessment. Classifiers are modeled as operations on features that can be measured and describe various aspects of potential threats, and each feature is assigned a cost that an adversary would incur in modifying that feature's measurement. Classifiers are designed by selecting features and applying one of several ML methods using a set of examples. One valuable lesson learned from this research is the suggestion that ensemble-based classifiers appear to be more resilient to adversarial actions than other classifier types. The paper has a fairly extensive bibliography referencing recent publications addressing the need to design classifiers for cybersecurity applications that take explicit account of the potential disruptions of adversarial activities.

Two feature selection algorithms are described: the adversary resilient algorithm and the k-relaxed feature selection algorithm. Adversary resilient algorithms utilize only features where an adversary would incur an unacceptably high cost to modify it, modeled as a cost greater than a threshold parameter's value. The k-relaxed feature selection algorithm allows additional features to be utilized by the classifier as long as the combined cost incurred by an adversary to modify those features is no greater than the value of the parameter k. As expected, an optimal (non-resilient) algorithm will perform best if no adversarial action is present, whereas the k-relaxed feature selection algorithm will produce a classifier that performs less well, and the adversary resilient algorithm will produce a classifier that performs least well. However, the non-resilient algorithm produces classifiers that more quickly lose effectiveness due to increased adversarial action.

The authors utilized a reasonably up-to-date collection of executables collected during November 2015 from Malwr (https://malwr.com, which is now defunct) to train and evaluate classifiers. The data set included "4312 executable analysis reports, including

1110 malware and 3202 benign samples" (p. 423). Malwr's cloud-based malware detection service was based upon the Cuckoo sandbox for malware analysis ("Cuckoo Sandbox – Automated Malware Analysis" n.d.), which is an open source dynamic analysis platform. According to the authors of the present paper, the "framework is based on a virtual machine that is used to run the tested executable and a host machine that manages the analysis process and collects relevant sensor readings" (p. 423). The dataset was labeled using VirusTotal ("VirusTotal" n.d.), and "a process was deemed malicious if one or more anti-virus engines classified it as such" (p. 423). Malwr was a "powerful, free, independent and non-commercial service to the security community, independent of academic researchers" (p. 423). Some alternatives to Malwr appear to be available ("Malwr Alternatives and Similar Websites and Apps - AlternativeTo.Net." n.d.).

Costs of designing an exploit that avoided detection yet had the same key features of each malware example in the data set were estimated by four independent content experts, each with at least 10 years of experience in the field. The costs were on a scale of 1-5, "with one denoting a minimal amount of effort, and five meaning 'practically impossible'" (p. 424). A threshold cost of 3.75 was used, and features with detection avoidance strategies with costs above this value were deemed infeasible and therefore safe for adversary resilient (most pessimistic) classifiers to use. A total of 52 features were available for use by the classification algorithms.

The classifier design and analysis approach described in this paper ignored the potential of cross-feature interactions (or correlations); each feature was either utilized, or not, by a specific classifier. This ignores the potential for both improved adversarial resilience and performance gains that might be achieved by optimization over a mixed feature strategy—for example, utilizing partial least squares to reduce the dimension of the feature space. Although the outcomes of this research are somewhat expected (decreased sensitivity to adversarial actions when classifiers utilize a reduced set of features that are difficult for an adversary to exploit, with a concomitant performance penalty when no adversarial action is present), there is still value in the findings. The authors state:

> Multisensor fusion is the basis of most modern cyber defense systems in which a variety of sensors are deployed throughout the defending organization, and the sensors' readings are analyzed collectively in an attempt to identify attacks. … We also explored the inherent resilience of different classification algorithms. The random forest classifier demonstrated superior resilience in all cases, suggesting that ensemble based classifiers are inherently more resilient to adversarial attacks. … Our work suggests that ensemble based classifiers are inherently more resilient than simple classifiers. (p. 428)

Klarreich, Erica. 2016. "Learning Securely." *Communications of the ACM* 59 (11): 12–14.

Klarreich discusses the possibility of "adversarial ML" or poisoning an ML system so that it makes incorrect decisions or classifications. She points out that this can be done with little knowledge of the structure of the ML algorithm. Examples include work by N. Carlini at UC Berkeley, "who has crafted audio files that sound like white noise to humans, but like commands to speech recognition algorithms" (p. 12) and work by I. Goodfellow at OpenAI, who was an author of the 2013 paper that modified an image of a stop sign so that it was classified as a yield sign. She states that it also has been shown that inputs designed to fool one ML system can also fool others:

> In a paper posted online in May [2016], Goodfellow … [and his co-authors] showed that adversarial examples transfer across five of the most commonly used types of machine learning algorithms: neural networks, logistic regression, support vector machines, decision trees, and nearest neighbors. The team carried out 'black box' attacks … on classifiers hosted by Amazon and Google. They found after only 800 queries to each classifier, that they could create adversarial examples that fooled the two models 96% and 89% of the time, respectively. (p. 14).

See (Papernot, McDaniel, and Goodfellow 2016) for details; a second paper (Carlini and Wagner 2017) explores the construction of adversarial examples in greater detail. See also the paper by Yuan (Yuan, He, Zhu, and Li 2019).

As ML algorithms become better able to perform unsupervised ML and are used to implement systems that adapt dynamically to changing conditions—for example, as an adaptive intrusion detection system—fields related to construction of adversarial ML examples will most likely become critically important. An adaptive ML system designed to detect threats might be re-trained by incoming examples to ignore a specific class of threats, poisoning the system to allow malware to escape detection.

Yampolskiy, Roman V., and M. S. Spellchecker. 2016. "Artificial Intelligence Safety and Cybersecurity: A Timeline of AI Failures." *ArXiv:1610.07997 [Cs],* October.

Yampolskiy discusses artificial intelligence safety, emphasizing that "[f]ully autonomous machines can never be assumed to be safe" (p. 8). Numerous historical examples are provided. The paper is a warning that complex systems have vulnerabilities and can have unintended consequences.

4. DARPA's Automated "Capture the Flag" Experiment

The vendors' reports on cybersecurity issues highlight two points that should be common knowledge in the cybersecurity field. First, both the number of attempted attacks on computers and networks and the speeds of execution of attacks continue to increase. Several factors drive this, including the potential rewards for success, the rich environment in which tools and capabilities can be shared or rented (such as within the so-called "dark web"), automation (including ML and AI), and harnessing of large collections of resources (examples include bot-nets used for distributed denial of service attacks and hijacked cloud instances). Second, defenders must effectively utilize automation to keep up with the attackers.

CTF events have been utilized in professional and educational settings to teach cybersecurity techniques – both defensive and offensive – for many years. In one form, players both attack competitors' systems and defend their own systems, according to specified ground rules, and they amass points by achieving goals such as gaining access to or retrieving information from targets. DARPA organized, funded, and ran a Cyber Grand Challenge in 2014–2016 that pitted machines and algorithms, rather than human players, against each other to explore the potential of automated discovery of and defense against information system vulnerabilities.

> *The competition began in 2014 with 104 teams. Through a series of qualifying events that required teams to demonstrate technical excellence, the number of teams was narrowed to 28 for the main qualifying event held in June 2015. Seven teams advanced to the finals. They were:*
>
> *University of Idaho (System Name: Jima)*
>
> *University of California, Santa Barbara (System Name: Mechaphish)*
>
> *University of California, Berkeley (System Name: Galactica)*
>
> *Raytheon, Inc. (System Name: Rubeus)*
>
> *University of Virginia and Grammatech, Inc. (System Name: Xandra)*
>
> *ForAllSecure, Inc. (System Name: Mayhem)*
>
> *Disekt (System Name: Crspy)*
>
> *In August 2016 at DEF CON 24, the seven teams faced off in a final contest.*
>
> (Davidson 2017, p. 1) [minor reformatting was performed]

The DARPA Cyber Grand Challenge (CGC) is of interest for a couple of reasons. First, it is reasonably clear that cyber defense must, to a significant degree, be automated,

if for no other reason than to rapidly trigger and alert human response teams and to help to prioritize their responses to attacks. Without automation, the amount of information that must be monitored is overwhelming. Second, the CGC's CTF design is a good way to evaluate the relative merits of different technologies (or algorithms) and different mixes of these technologies in a somewhat realistic scenario. Defensive strategies such as automated mitigation or patching may be directly applicable, and although offensive strategies are unlikely to be of use in most situations, these strategies are the same strategies that may be utilized during software or system development to test and evaluate implementations.

DARPA stated its justification of the CGC as follows:

> The need for automated, scalable, machine-speed vulnerability detection and patching is large and growing fast as more and more systems—from household appliances to major military platforms—get connected to and become dependent upon the internet. Today, the process of finding and countering bugs, hacks, and other cyber infection vectors is still effectively artisanal. Professional bug hunters, security coders, and other security pros work tremendous hours, searching millions of lines of code to find and fix vulnerabilities that could be taken advantage of by users with ulterior motives. (Fraze n.d., p. 1)

Anticipated future benefits of the CGC listed on this web page included:

- "Expert-level software security analysis and remediation, at machine speeds on enterprise scales,
- Establishment of a lasting R&D community for automated cyber defense, and
- Creation of a public, high-fidelity recording of real-time competition between automated cyber defense systems." (p. 1)

Mayhem, the system developed by ForAllSecure, Inc. won the DARPA CGC; the strategy is partially described by Brumley (2019). AI and ML are described only once in the last paragraph of the article:

> Right now, ForAllSecure is selling the first versions of its new service to early adopters, including the U.S. government and companies in the high-tech and aerospace industries. At this stage, the service mostly indicates problems that human experts then go in and fix. For a good while to come, systems like Mayhem will work together with human security experts to make the world's software safer. In the more distance future, we believe that machine intelligence will handle the job alone. (p. 35)

The technology used by ForAllSecure appears to depend upon work performed by three of its founders (Avgerinos, Brumley, and Rebert) and S. K. Cha at Carnegie Mellon University and described in four U. S. Patents (Brumley, Cha, and Avgerinos 2015; Brumley, Cha, Avgerinos, and Rebert 2015; Avgerinos, Rebert, and Brumley 2017; Brumley, Cha, Avgerinos, and Rebert 2017), cited in order of their filing dates. None of

these patents mentions artificial intelligence or ML. The specifications of these patents indicate that the methods utilize symbolic execution of portions of target software (which may be source code, an intermediate representation such as a byte code, or machine level instructions) combined with logic (using function representations of the portion's actions) to limit the size of the search space and heuristics to prioritize searches to find vulnerabilities in the software. Both dynamic and static symbolic execution can be used; a bibliography of papers related to symbolic execution is available on GitHub (Anand n.d.). A module is then utilized to generate code to exploit each generated vulnerability, which is subsequently tested using a verification module. The Mayhem system subsequently generates a patch, which must also be tested and then applied to its executing software to protect itself against the competitors in the competition. The most recent patent (Brumley, Cha, Avgerinos, and Rebert 2017) appears to describe aspects of the Mayhem system.

The Mayhem system implemented by ForAllSecure thus appears to utilize symbolic execution of target software combined with a heuristic search strategy and a substantial amount of parallelization (using the hardware provided by DARPA for the CGC). Although heuristic search is also a feature of AI systems, this does not mean Mayhem qualifies as an AI system (though it does not exclude the possibility). The other popular method used to find vulnerabilities in software is fuzzing, which exercises the target software using randomly generated inputs. Vulnerabilities are found when an input causes the software to misbehave or crash. An open source project that implements fuzzing is American Fuzzy Lop (Zalewski n.d.), which also uses various heuristics to attempt to more quickly find vulnerabilities.

Xandra, the system developed by GrammaTech and the University of Virginia, placed second in the DARPA CGC. Xandra utilized "fuzzing pods capable of 1.8M fuzzing ops per second" and binary code analyzers, combined with patch generators to repair binaries. ("GrammaTech's Team TECHx Places Second in DARPA's Cyber Grand Challenge." n.d., p. 1). According to an article in Wikipedia (not considered authoritative), GrammaTech is a spin-off from Cornell University, and their current research focuses on static and dynamic analysis of source code and binaries. ("GrammaTech" 2019). Their products include CodeSonar ("CodeSonar" 2015) and CodeSonar/x86 ("Binary Static Analysis with CodeSonar" 2015).

Mechaphish (Mechanical Phish), developed by the team from the University of California, Santa Barbara, placed third in the DARPA CGC. The team, now known as Shellphish, has published their software as open source. The software relies in part upon binary code analysis using a framework called "angr" ("DARPA CGC ~ Shellphish" n.d.). The research group has published several papers that document their work. ("Cyber Grand Shellphish" n.d.; Shoshitaishvili et al. 2016, 2018; Stephens et al. 2016; Wang et al. 2017;) The Driller component of Mechaphish (Stephens et al. 2016) is described as an augmentation of fuzzing methods using selective concolic execution to enable deeper

exploration of target software. The term "concolic" execution refers to a combination of concrete and symbolic execution, where variables are selectively instantiated to concrete, or non-symbolic, values. See "Concolic Testing" (2019) for a brief discussion of concolic testing and several references, including some references to commercial systems.

Rubeus, the system developed by Raytheon, placed fourth in the DARPA CGC. ("Raytheon: The Bot Defenders — Humans and Machines Team up to Defeat Cyber Attacks" n.d.). Rubeus utilized static analysis tools and symbolic execution, and a cyber reasoning system that "used advanced analytics, autonomous reverse engineering software, and continuous machine learning." ("Cyber-Physical Systems and Autonomy — Highlighting Raytheon Company's Work" 2017, p. 2).

Galactica, the system developed by CodeJitsu at the University of California, Berkeley, CA, Syracuse, NY, and Lausanne, Switzerland, placed fifth in the DARPA CGC. Galactica utilized coverage-based "graybox" fuzzing based upon AFLfast, a fork of the AFL open source project described above (Zalewski n.d.). The technology is described by Böhme (Böhme, Pham, and Roychoudhry 2018), and the AFLfast software is available on GitHub (Böhme n.d.). Although Galactica placed fifth in the CGC, it placed second in the number of bugs found.

Jima, the system developed by the Center for Secure and Dependable Systems (CSDS) at the University of Idaho in Moscow, ID, placed sixth in the DARPA CGC. The system was developed by a two-person team (Drs. Jia Song and Jim Alves-Foss) and utilized a black-box fuzzing approach to detect vulnerabilities and static analysis of the target binaries. Some documentation of the techniques used is available in two published articles, but both were written before the final competition. (Song and Alves-Foss 2015, 2016).

Crspy, the system developed by Disekt at the University of Georgia in Athens, GA, placed seventh in the DARPA CGC. No published information was located that describes this system.

The following bullets summarize findings from this evaluation of the published artifacts from the DARPA Cyber Grand Challenge:

- Automated discovery and correction of vulnerabilities (or defects of specific types) in software is not only possible; it has been demonstrated.
- Automated discovery of vulnerabilities can be accomplished using fuzzing. See, for example, the listing of vulnerabilities discovered using AFL at (Zalewski n.d.). See also the AFLfast fork of AFL at (Böhme n.d.; Böhme, Pham, and Roychoudhry 2018).
- Static software analysis and dynamic execution of software (using instrumentation, symbolic execution, or a combination of the two) are also viable technologies for vulnerability discovery.

- Symbolic execution provides the opportunity to utilize automated theorem provers to test a vulnerability (proof by construction) and guide automated code patching.
- These methods can be applied to either source code or binary executables and should be independent of the implementation language.
- Open source tools are available, and some commercial products are either available or under development.
- The value of these technologies is two-fold. First, they can be incorporated in the testing (verification and validation) phases of software life cycle processes. Second, they offer ways to test and patch third-party software that is being incorporated into another system or software during development.
- The DARPA Cyber Grand Challenge's primary focus was upon cybersecurity, and these technologies not only have obvious utility in this context but are also in active use. However, the same technologies have the potential for much broader applicability to software and system testing, and there are indications that they are being adopted.

DARPA held a proposers' day for a new program on April 19, 2018. The program is called Computers and Humans Exploring Software Security (CHESS) and is managed from the DARPA Information Innovation Office (I2O); a broad agency announcement (BAA – funding opportunity number HR001118S0040)[12] was released on April 18, 2018. This program is a logical successor to the DARPA Cyber Grand Challenge. The BAA states that the "goal of the CHESS program is to develop computer-human systems to rapidly discover all classes of vulnerability in complex software. These novel approaches for the rapid detection of vulnerabilities will focus on identification of system information gaps that require human assistance, generation of representations of these gaps appropriate for human collaborators, capture and integration of human insights into the analysis process, and the synthesis of software patches based on this collaborative analysis" ("Computers and Humans Exploring Software Security (CHESS)." Broad Agency Announcement HR001118S0040 2018, p. 4).

The introductory section of the BAA makes clear that the automated vulnerability analysis and patching systems of the DARPA Cyber Grand Challenge are not sufficient. According to the BAA, automated tools must (a) be able to find and reason about significantly more types of vulnerabilities than were addressed by prior automation and (b) be able to collaborate with human analysts to address these vulnerabilities. These goals can be interpreted as a desire for a more automated and "intelligent" security information and event managers SIEM. Quoting from the introduction:

[12] See "Computers and Humans Exploring Software Security (CHESS)." Broad Agency Announcement HR001118S0040. DARPA. April 18, 2018, and "Computers and Humans Exploring Software Security (CHESS) - DARPA-SN-18-40 (Archived) - Federal Business Opportunities: Opportunities" n.d.

> The Department of Defense (DoD) maintains information systems that depend on Commercial off-the-shelf (COTS) software, Government off-the-shelf (GOTS) software, and Free and open- source (FOSS) software. Securing this diverse technology base requires highly skilled hackers who reason about the functionality of software and identify novel vulnerabilities. This process requires hundreds or thousands of hours of manual effort per discovered vulnerability and does not scale sufficiently to secure the continuously growing technology base.
>
> Hackers use program analysis techniques and tools to identify and mitigate vulnerabilities, but this process requires considerable expertise, manual effort, and time. These techniques include dynamic analysis, static analysis, symbolic execution, constraint solving, data flow tracking, and fuzz testing. Automated program analysis capabilities can reason over only a few vulnerability classes without human involvement, such as memory corruption or integer overflow, but cannot address the majority of vulnerabilities. These unaddressed vulnerability types depend on subtle semantic and contextual information, which is beyond the grasp of modern automation. Scaling up existing approaches to address the size and complexity of modern software packages is not possible given the limited number of expert hackers in the world, much less the DoD.
>
> The CHESS program will develop capabilities to discover and address vulnerabilities of all types in a scalable, timely, and consistent manner. DARPA believes that achieving the necessary scale and timelines in vulnerability discovery will require innovative combinations of automated program analysis techniques with support for advanced computer-human collaboration (CHC). Due to the cost/scarcity of expert hackers, such capabilities must be able to collaborate with humans of varying skill levels, even those with no previous hacking experience or relevant domain knowledge. (pp. 4-5)

The BAA includes Figure 1 (p. 6) to illustrate the CHESS concept. The blue boxes in the figure represent the automation, which is to be used to discover vulnerabilities in either source code or binaries and generate a representation of the discovered information and provide context for human analysts. Proof of detected vulnerabilities may be a collaborative activity involving both automation and humans.

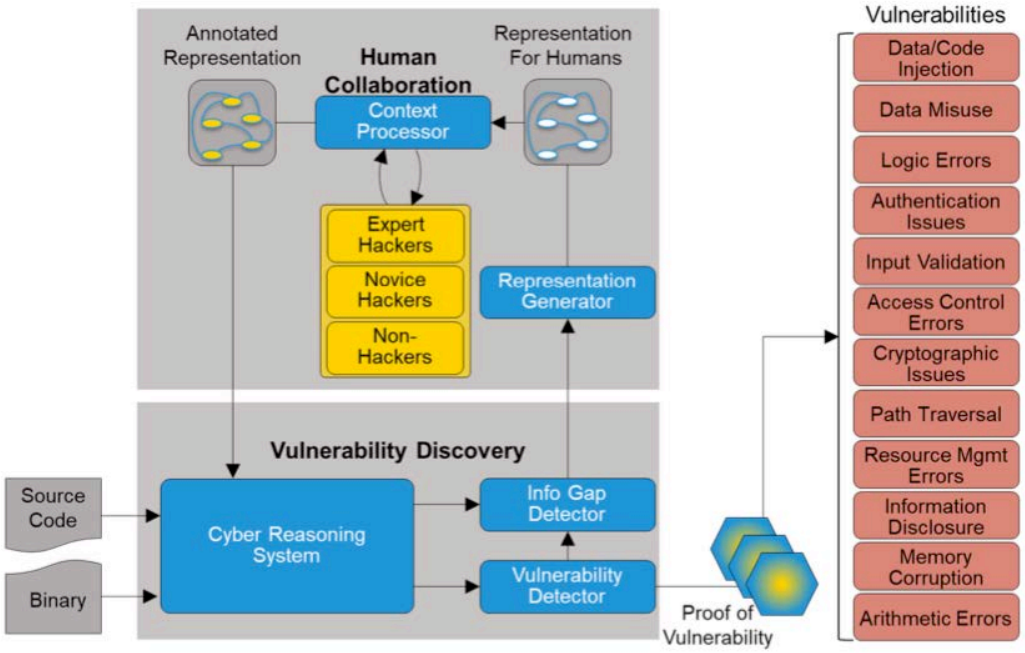

Figure 1. CHESS System Overview.

Proposers were allowed to suggest work in five technical areas: human collaboration, vulnerability discovery, "voice of the offense," control team, and integration, test, and evaluation (p. 8). The proposed work was to be applicable to at least one of the C/C++, Python, and JavaScript programming languages and either the Linux or Windows operating system environment. The CHESS program is designed to target the vulnerability classes listed in Table 1 (p. 7), using the Common Weakness Enumerations (CWEs) defined in MITRE's CWE List Version 3.0.[13]

Vulnerability Class	Parent CWEs
Data/Code Injection	74
Data Misuse	471, 501, 610, 628, 642, 662, 665, 673, 704, 706
Logic Errors	691, 697, 703, 758, 768
Authentication Issues	287
Input Validation	20, 138, 170, 172, 228, 463
Access Control Errors	269, 285, 282, 286, 923
Cryptographic Issues	324, 325, 326, 330, 347
Path Traversal	22, 41, 59
Resource Management Errors	400, 404, 405, 665, 666
Information Disclosure	668
Memory Corruption	118
Arithmetic Errors	682

Table 1. Target Vulnerability Classes.

[13] See https://cwe.mitre.org/data/index.html.

Recent announcements indicate that DARPA has awarded at least one contract to a group that includes members of the Shellphish team. ("Bianchi Co-PI in $11.7 Million DARPA CHESS Grant | Department of Computer Science" n.d.; "CHECKMATE! SecLab Receives $11.7 Million Grant from DARPA CHESS Program | UCSB Computer Science" n.d.)

Two other DARPA programs are of possible interest for automated testing of software: the VET program ("Vetting Commodity IT Software and Firmware (VET)" n.d.) and the APAC program ("Automated Program Analysis for Cybersecurity (APAC)" n.d.). According to the archived information about the VET program,

> DARPA created the Vetting Commodity IT Software and Firmware (VET) program to address the threat of hidden malicious functionality in COTS IT devices. VET's goal is to demonstrate that it is technically feasible to determine that the software and firmware shipped on commodity IT devices is free of broad classes of hidden malicious functionality. ("Vetting Commodity IT Software and Firmware (VET)" n.d., p. 1).

DARPA's description of the APAC program follows:

> The Automated Program Analysis for Cybersecurity (APAC) program aims to address the challenge of timely and robust security validation of mobile apps by first defining security properties to be measured against and then developing automated tools to perform the measuring. APAC will draw heavily from the field of formal-methods program analysis (theorem proving, logic and machine proofing) to keep malicious code out of DoD Android-based application marketplaces. ("Automated Program Analysis for Cybersecurity (APAC)" n.d., p. 1).

Thus, the VET program focuses upon commodity hardware, while the APAC program's focus is upon mobile apps. Both of these programs warrant additional exploration to determine their relevance to the current task.

5. Books and Reports Covering AI and ML in the Context of Cybersecurity

Bowen, Pauline, Joan Hash, and Mark Wilson. 2007. "Information Security Handbook: A Guide for Managers." *NIST Special Publication (SP) 800-100*. Gaithersburg, MD: National Institute of Standards and Technology.

This handbook is part of the *NIST Special Publication 800* series of documents relating to information system security and provides high-level guidance for managers. It is referenced here as an overview of the various issues surrounding information system security, and particularly, cybersecurity. From the Introduction:

> This Information Security Handbook provides a broad overview of information security program elements to assist managers in understanding how to establish and implement an information security program. … The purpose of this publication is to inform members of the information security management team (agency heads; chief information officers [CIOs]; senior agency information security officers [SAISOs], also commonly referred to as Chief Information Security Officers [CISOs]; and security managers) about various aspects of information security that they will be expected to implement and oversee in their respective organizations. In addition, the handbook provides guidance for facilitating a more consistent approach to information security programs across the federal government. (p. 1).

The handbook provides an overview of the system development life cycle and includes chapters on security awareness and training, capital planning and investment, interconnecting systems, performance metrics, security planning including personnel roles and responsibilities and the system security plan, contingency planning, risk management, certification, accreditation, security assessments, security services and products, incident response, and configuration management.

Chio, Clarence, and David Freeman. 2018. *Machine Learning and Security: Protecting Systems with Data and Algorithms*. O'Reilly Media, Inc.

The publisher's description[14] of the book begins with two questions:

> Can machine learning techniques solve our computer security problems and finally put an end to the cat-and-mouse game between attackers and defenders? Or is this hope merely hype? Now you can dive into the science and answer this question for yourself. With this practical guide, you'll

[14] See https://learning.oreilly.com/library/view/machine-learning-and/9781491979891/#toc.

explore ways to apply machine learning to security issues such as intrusion detection, malware classification, and network analysis.

The first chapter addresses the question: "Why machine learning and security?" It provides a discussion of the cyber attacker's economy and the marketplace for hacker skills, and some brief comments on how adversaries are using ML. The chapter concludes with an example that constructs a spam filter using the Natural Language Toolkit (NLTK)[15] in Python and the 2007 TREC Public Spam Corpus data set[16]. An iterative approach is used that adds a hashing scheme to generate signatures and achieves an 88.6% classification accuracy on the test data set.

The second chapter provides an overview of classification methods, including the logistic regression, DT, decision forest, SVM, naïve Bayes, KNN, and neural network methods of supervised classification, followed by a discussion of training set construction, feature selection, over/under fitting, and receiver operating characteristic (ROC) curves. Clustering methods are then discussed, including k-means, hierarchical clustering, locally sensitive hashing, k-d trees, and density-based spatial clustering of applications with noise (DBSCAN). A good bibliography is included.

Subsequent chapters discuss anomaly detection in intrusion detection systems, malware analysis, network traffic analysis, protecting the consumer web, production systems (scalability), and adversarial ML. The book's examples assume some knowledge of programming languages, with examples that use Python, C, and Java, and some knowledge of assembly language code and executable file formats, as well as an acquaintance with web and other network protocols, UNIX shell commands and a debugger. The chapter on network traffic analysis assumes some knowledge of firewalls or other packet filtering technologies.

The book provides an excellent overview of several aspects of both the problems faced by those charged with protecting cyber infrastructure and of the technologies that can be used to address these problems. The book contains sufficient details to gain an understanding of the methods that are discussed with references to the literature for more details.

Fazeldehkordi, E., O. A. Akanbi, and Iraj Sadegh Amiri. 2014. *A Machine-Learning Approach to Phishing Detection and Defense*. Syngres.

This publication is a research report that compares the performance of four types of classifiers—DTs, KNN (with multiple values of k), SVM, and linear regression—as they

[15] See https://www.nltk.org/.

[16] See https://plg.uwaterloo.ca/~gvcormac/spam/. See also the NIST web site that describes the Text Retrieval Conference (TREC) series and available data: https://trec.nist.gov/.

are used to detect phishing web sites. Performance was assessed using accuracy and FARs on a validation subset of the collected data. The data set used to train, test, and validate the classifiers was constructed using 3,611 phishing web sites collected during 2008–2012 and documented in the Phishtank open source repository[17] and an additional 1,638 non-phishing web sites. Only web sites that were active at the time of the study were included. This dataset is divided into three parts for training, testing, and validation.

The report provides a good description of the process used to construct and test classifiers for phishing web sites, including information about features extracted from the web sites and normalization of the feature data prior to building classifiers. Features include information extracted from the URL of each web site such as URL length, whether a numeric or hexadecimal IP address is embedded in the URL, whether encryption is utilized (HTTPS), and the number of periods present in the URL. Features from the web site include the presence of embedded forms with submit buttons, the presence of links that include an "at" symbol, and the presence of empty pages.

The reported accuracies of the constructed classifiers were above 98.5% with low FARs in all cases. However, there were some flaws in the research that call these results into question. Primary among these is the failure to choose multiple randomly selected, disjoint subsets of the data set for training and test and report results over multiple runs, a lack of a description of how non-phishing web sites were selected, and the small size of the data set. The authors acknowledged the limitations due to the small size of the data set.

Although the report does not provide classifiers that could be generalized and used in realistic scenarios, its description of the data set curation, feature extraction, and classifier generation and test processes should be valuable to a reader interested in learning more about the technology.

Gil, Laurent, and Allan Liska. 2019. *Security with AI and Machine Learning*. Sebastopol, CA: O'Reilly Media, Inc.

The publisher's description of this book[18] states: "For security professionals seeking reliable ways to combat persistent threats to their networks, there's encouraging news. Tools that employ AI and ML have begun to replace the older rules- and signature-based tools that can no longer combat today's sophisticated attacks. In this ebook, Oracle's Laurent Gil and Recorded Future's Allan Liska look at the strengths (and limitations) of AI- and ML-based security tools for dealing with today's threat landscape. This high-level overview demonstrates how these new tools use AI and ML to quickly identify threats, connect attack patterns, and allow operators and analysts to focus on their core mission.

[17] See https://www.phishtank.com/.

[18] See https://learning.oreilly.com/library/view/security-with-ai/9781492043133/.

You'll also learn how managed security service providers (MSSPs) use AI and ML to identify patterns from across their customer base."

The focus of the book is upon threats posed by bots and botnets. In chapter 3, a bot is defined as "a piece of code that automates and amplifies the ability of an attacker to exploit as many targets as possible as quickly as possible." Botnets are distributed collections of computer processors running bots that operate in a coordinated fashion to perform attacks. An example is a botnet used to perform a distributed denial of service (DDoS) attack by flooding a site with network traffic. Other botnets implement online fraud schemes and ransomware used to encrypt information in a target's system and extort funds from the victim. The book discusses the three functions of bots and botnets: scanning, exploitation, and command, communications, and control. The use of AI and ML to detect anomalies in network traffic is discussed as a method of countering the threats posed by bots and botnets.

The authors discuss the potential use of AI and ML in other security applications, including identification of insider threats via "user and entity behavior analytics," security information and event managers (SIEMs), and next generation anti-virus products.

Unfortunately, although this book does provide a very high-level description of uses and opportunities for future use of AI and ML in cyber security, few specifics are offered. The book may serve as a quick introduction to cyber security as it relates to bots and botnets, but a reader who is looking for details about how AI and ML are used should look elsewhere.

Cylance Data Team. 2017. *Introduction to Artificial Intelligence for Security Professionals*. Irvine, CA: The Cylance Press.

This book provides an excellent introduction to clustering, classification, probability, and deep learning, with examples and accompanying sample software and data sets.[19] The authors describe two algorithms for each topic in sufficient detail that the reader can implement and test them for simple scenarios.

The k-means and DBSCAN clustering algorithms are described, along with discussions of "data selection and sampling, feature extraction, feature encoding and vectorization, model computation and graphing, and model validation and testing" (p. xvi). A discussion of principal component analysis (PCA) for data reduction is included in the description of the k-means algorithm—a critically important algorithm in ML that is not described in most of the other references reviewed in this report. A "hands-on learning section showing how *k*-means and DBSCAN models can be applied to identify exploits similar to those associated with the Panama Papers breach" (p. xvii) is used to illustrate the algorithms.

[19] See https://www.cylance.com/intro-to-ai.

Logistic regression and the CART decision tree algorithms are described in the chapter on classification, which also includes a discussion of differences between supervised and unsupervised learning and between linear and non-linear classifiers. The model training, validation, testing, and deployment phases of classifier construction are covered. The discussion of logistic regression includes concepts such as regression weights, regularization and penalty parameters, decision boundaries, and fitting data, and the discussion of the CART algorithm includes node types, split variables, benefit scores, and stopping criteria. Assessment and validation using confusion matrices and precision and recall metrics are also covered. The hands-on learning material shows "how logistic regression and decision tree models can be applied to detect botnet command and control systems that are still in the wild today" (p. xviii).

The chapter on probability discusses its use for predictive modeling and includes descriptions of naïve Bayes classification and Gaussian mixture model clustering. Concepts including random trials, outcomes, and events; joint and conditional probabilities; prior and posterior probability; the Gaussian (normal) density; expectations; and maximum likelihood techniques (called "expectation maximization optimization") are discussed. A small flaw in the treatment of these subjects is the use of the term "likelihood," which is a synonym for "probability" in this context; methods that utilize likelihood ratios do not appear to be covered. The hands-on learning material shows "how [naïve Bayes] and [Gaussian mixture] models can be applied to detect spam messages sent via [short message system (SMS)] text" (p. xviii).

Convolutional neural networks and long short-term memory are covered in the chapter on deep learning. Concepts discussed include feedforward and recurrent neural networks, and neural network "nodes, hidden layers, hidden states, activation functions, context, learning rates, dropout regularization, and increasing levels of abstraction" (p. xix). The chapter concludes with a demonstration of "how LSTM [long short-term memory] and CNN models can be applied to determine the length of the XOR key used to obfuscate a sample of text" (p. xix).

Kissel, Richard, Kevin Stine, Matthew Scholl, Hart Rossman, Jim Fahlsing, and Jessica Gulick. 2008. "Security Considerations in the System Development Life Cycle." *NIST Special Publication (SP) 800-64 Rev. 2*. Gaithersburg, MD: National Institute of Standards and Technology.

The Executive Summary in this publication states that it "has been developed to assist federal government agencies in integrating essential information technology (IT) security steps into their established IT system development life cycle (SDLC). This guideline applies to all federal IT systems other than national security systems. The document is intended as a reference resource rather than as a tutorial and should be used in conjunction

with other NIST publications as needed throughout the development of the system. … To be most effective, information security must be integrated into the SDLC from system inception" (p. 1). The document assumes a waterfall SDLC process, but the information should be easily adaptable to other process models.

The guide provides an overview of key security roles and responsibilities necessary in most software life cycle processes. An overview of the (waterfall) SDLC follows. Chapter 3 provides guidance on incorporation of security into each phase of the SDLC, and Chapter 4 "highlights security considerations for development scenarios, such as service-oriented architectures and virtualization, for which the approach to security integration varies somewhat from that of traditional system development efforts" (p. 3).

Several of the appendices in this report should prove valuable. Appendix D maps NIST publications to SDLC activities. Appendix E provides an overview of other SDLC methodologies. Appendix G presents a graphical view of security integration into a SDLC.

Pino, Robinson E., Alexander Kott, and Michael Shevenell, (Eds). 2014. *Cybersecurity Systems for Human Cognition Augmentation*. Advances in Information Security 61. New York, NY: Springer.

This book is an edited volume of 13 chapters, each written by a different set of authors. Thus, it represents a snapshot of research efforts circa 2014 that relate emerging fields of AI and ML to cybersecurity. One emphasis of the collection of papers is the emerging field of neuromorphic computing (NC).[20] NC researchers have proposed several novel computing architectures inspired by biological neural tissues in the brain. NC architectures can operate using spiking signals rather than continuous or logic level-based signals, and thus hold the potential for extremely low power operation. The novel architectures also offer the potential for doing more with less—implementing useful systems using less complex hardware and algorithms than, for example, systems based upon more traditional neural networks and deep learning methods. A second emphasis in this collection is the use of novel semiconductor components, such as memristors, that exhibit programmable variable resistivity.

Several chapters in this work are of potential interest. Chapter 4 discusses classifier methodologies for use in intrusion detection systems, including KNN, SVMs, ANNs, self-organizing maps, DTs, naïve Bayes, genetic algorithms, and fuzzy logic. Other topics discussed in this chapter include cyber situational awareness and malicious code detection. However, although the listed references are of use, the discussion is at a fairly high level and is light on details.

[20] For a recent survey of the neuromorphic computing literature, including literature describing the implementation of neural networks in hardware, see (Schuman, Catherine D. et al. 2017).

Chapter 7 discusses the use of neural network-based malware detection on Android mobile devices. Detection systems based upon an app's use of permissions and upon system call patterns are compared, with some experimental results. This chapter's bibliography provides pointers to other works in this field.

Chapters 9–12 discuss various properties and uses of memristor devices in ANNs and neuromorphic architectures, with some discussion of their potential use in intrusion detection systems.

Chapter 13 discusses cyber security issues for systems with reconfigurable hardware and for embedded systems, as well as the possibility of designing such systems for resilience.

This book provides perspectives on the potential of emerging technologies but most likely does not provide information that would prove useful in the near to medium term. It should be noted, however, that adoption of emerging technologies may become essential in some scenarios because of continuing increases in communications data rates or power limitations.

The first-listed editor of this volume, Dr. Robinson E. Pino, is at present the Director (Acting) of the Research Division of Advanced Scientific Computing Research (ASCR) at the Department of Energy.

Scarfone, Karen and Peter Mell. 2012. "Guide to Intrusion Detection and Prevention Systems (IDPS)." *NIST Special Publication (SP) 800-94 Rev. 1* (Draft). Gaithersburg, MD: National Institute of Standards and Technology.

This is a draft report that provides a guide to intrusion detection and prevention systems (IDPS). This is a 2012 revision in progress of the first edition of *NIST Special Publication 800-94*, dated February 2007. These comments apply to both versions. Chapter 2 discusses principles of IDPS, covering signature-based and anomaly-based detection, as well as stateful protocol analysis. Chapter 3 provides an overview of IDPS technologies, including the components and architectures, security capabilities, and management of these technologies. Chapters 4 and 5 provide overviews of network-based and wireless IDPS, respectively, and Chapters 6 and 7 discuss network behavior analysis systems and host-based IDPS. The use and integration of multiple IDPS technologies is discussed in Chapter 8, while Chapter 9 provides guidance on IDPS product selection.

Tamburello, Paul, and Peter Guerra. 2018. *Modernizing Cybersecurity Operations with Machine Intelligence*. Sebastopol, CA: O'Reilly Media, Inc.

This book provides a very high-level view of the potential utility of machine intelligence in cybersecurity. The book is worth reviewing but is not detailed. It is not an unbiased or critical review of this area.

6. Summary

This work and paper partly fulfill the following paragraphs of the statement of work in the project description: (3g), which states the intent to "evaluate technical options and alternatives … for standing up an enterprise-level Army Application Development Environment (ADE) that supports development for the full range of software platforms…"; (3j), which states the intent to "investigate options for automating the application vetting process using commercial workflow tools and software testing best practices"; and deliverable (4d), "a draft report on maturity and applicability of options that can support the creation of an Army ADE."

This paper provides a partial response to the question: "Do technologies related to AI and ML provide opportunities to improve the cybersecurity of SDLEs and their products?" This question was motivated in part by the perception that automation plays a very significant role in the "dark side" of cybersecurity as tools used to exploit information systems and organizations, compromise their functions, and exfiltrate their information. The time delay from a successful exploit to utilization of the compromised system or information can be extraordinarily short. It has become imperative that organizations that manage SDLEs achieve correspondingly short threat response time delays.

The report discusses selected publications that relate AI or ML to cybersecurity, and specifically to the context of cybersecurity of SDLE and their products. Publications were selected for review using a variety of sources, as described in the Executive Summary. The cited publications are not necessarily good sources of information. The report's summary of each publication can be viewed as an indication of merit as well as content, allowing the reader to allocate time to those of most interest while avoiding those from which little insight might be gained. This may be of particular use with the cited books.

There is an immense body of published work relating to AI and ML that extends over at least six decades. Although the corpus of literature relating to cybersecurity has a shorter time line (about three decades), there can be no claim that a brief review such as was undertaken here can be authoritative. Nevertheless, the authors hope that the discussions and observations that are offered can assist in determinations and selections of technologies, products, and methods for incorporation in high-quality SDLEs.

7. References

"A.M. Turing Award." n.d. Accessed April 9, 2019. https://amturing.acm.org/.

Abu-Ghazaleh, N., D. Ponomarev, and D. Evtyushkin. 2019. "How the Spectre and Meltdown Hacks Really Worked." *IEEE Spectrum* 56 (3): 42–49. https://doi.org/10.1109/MSPEC.2019.8651934.

Anand, Saswat. n.d. *A Bibliography of Papers Related to Symbolic Execution: Saswatanand/Symexbib*. https://github.com/saswatanand/symexbib. Accessed March 24, 2019.

Amit, Idan, John Matherly, William Hewlett, Zhi Xu, Yinnon Meshi, and Yigal Weinberger. 2018. "Machine Learning in Cyber-Security - Problems, Challenges and Data Sets." *ArXiv:1812.07858 [Cs, Stat]*, December. http://arxiv.org/abs/1812.07858.

"Automated Program Analysis for Cybersecurity (APAC)." n.d. Accessed March 25, 2019. https://www.darpa.mil/program/automated-program-analysis-for-cybersecurity.

Avgerinos, Thanassis, Alexandre Rebert, and David Brumley. 2017. Methods and systems for automatically testing software. United States Patent 9,619,375, filed May 21, 2015, and issued April 11, 2017. https://patents.google.com/patent/US9619375.

Bem, Jeremy, Georges R. Harik, Joshua L. Levenberg, Noam Shazeer, and Simon Tong. 2007. Large scale machine learning systems and methods. United States Patent 7,222,127, filed December 15, 2003, and issued May 22, 2007. https://patents.google.com/patent/US7222127B1/en.

Bertsekas, D. P. 1999. *Nonlinear Programming*. 2nd ed. Athena Scientific.

"Bianchi Co-PI in $11.7 Million DARPA CHESS Grant | Department of Computer Science." n.d. Accessed March 25, 2019. https://cs.uiowa.edu/resources/news/bianchi-co-pi-117-million-darpa-chess-grant.

"Binary Static Analysis with CodeSonar." 2015. Text. Binary Code Static Analysis with CodeSonar. October 9, 2015. https://www.grammatech.com/products/binary-analysis.

Birdwell, J. D., and R. D. Horn. 1990. "Optimal Filters for Attribute Generation and Machine Learning." In *29th IEEE Conference on Decision and Control*, 1537–39 vol.3. https://doi.org/10.1109/CDC.1990.203869.

Birdwell, J Douglas, Roger D Horn, and Shent Liang. 1992. "Automatic Generation of Signal Classification Algorithms Using Machine Learning." In *Recent Advances in Computer Aided Control Systems Engineering*, edited by M Jamshidi and C J Herget, 192–219. Studies in Automation and Control 9. Elsevier.

Böhme, M., V. Pham, and A. Roychoudhury. 2018. "Coverage-Based Greybox Fuzzing as Markov Chain." *IEEE Transactions on Software Engineering*, 1–1. https://doi.org/10.1109/TSE.2017.2785841.

Böhme, Marcel. n.d. *AFLFast (Extends AFL with Power Schedules). Accessed March 25, 2019.* https://github.com/mboehme/aflfast.

Bowen, Pauline, Joan Hash, and Mark Wilson. 2007. "Information Security Handbook: A Guide for Managers." NIST Special Publication (SP) 800-100. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-100.

Brumley, David. 2019. “The White-Hat Hacking Machine: How Mayhem Won a DARPA Challenge to Find and Patch Software Vulnerabilities.” *IEEE Spectrum* 56 (2): 30–35.

Brumley, David, Sang Kil Cha, Thanassis Avgerinos, and Alexandre Rebert. 2017. Detecting exploitable bugs in binary code. United States Patent 9,542,559, filed August 17, 2015, and issued January 10, 2017. https://patents.google.com/patent/US9542559B2.

Brundage, Miles et al. 2018. “The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation.” *ArXiv:1802.07228 [Cs]*, February. http://arxiv.org/abs/1802.07228.

Buczak, A. L., and E. Guven. 2016. “A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection.” *IEEE Communications Surveys Tutorials* 18 (2): 1153–76. https://doi.org/10.1109/COMST.2015.2494502.

Carlini, N., and D. Wagner. 2017. “Towards Evaluating the Robustness of Neural Networks.” In *2017 IEEE Symposium on Security and Privacy (SP)*, 39–57. https://doi.org/10.1109/SP.2017.49.

“CHECKMATE! SecLab Receives $11.7 Million Grant from DARPA CHESS Program | UCSB Computer Science.” n.d. Accessed March 25, 2019. https://www.cs.ucsb.edu/news/3341.

Chio, Clarence, and David Freeman. 2018. *Machine Learning and Security: Protecting Systems with Data and Algorithms*. O’Reilly Media, Inc.

Ciccatelli, Amanda. 2016. “Will Artificial Intelligence Revolutionize Cybersecurity?” *Inside Counsel Breaking News*, June 27, 2016. General OneFile. http%3A%2F%2Flink.galegroup.com%2Fapps%2Fdoc%2FA456285758%2FITOF%3Fu%3Dtel_a_utl%26sid%3DITOF%26xid%3D0ab8c13f.

“Cisco 2018 Annual Cybersecurity Report.” 2018. San Jose, CA: Cisco Systems, Inc.

“CodeSonar.” 2015. Text. GrammaTech. October 5, 2015. https://www.grammatech.com/products/codesonar.

“Computers and Humans Exploring Software Security (CHESS) - DARPA-SN-18-40 (Archived) - Federal Business Opportunities: Opportunities.” n.d. Accessed March 25, 2019. https://www.fbo.gov/index?s=opportunity&mode=form&id=557cfe6440e774224a008f6923e526f3&tab=core&_cview=0.

“Computers and Humans Exploring Software Security (CHESS).” Broad Agency Announcement HR001118S0040. DARPA. April 18, 2018.

“Concolic Testing.” 2019. In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Concolic_testing&oldid=879901800.

“CrowdStrike 2018 Global Threat Report.” n.d. PathFactory. Accessed March 24, 2019. https://crowdstrike.lookbookhq.com/global-threat-report-2018-web/cs-2018-global-threat-report?utm_campaign=Threat_Report&utm_medium=Web&utm_source=Marketo.

“Cuckoo Sandbox - Automated Malware Analysis.” n.d. Accessed April 9, 2019. https://cuckoosandbox.org/.

“Cyber Grand Shellphish.” n.d. Accessed March 24, 2019. http://www.phrack.org/papers/cyber_grand_shellphish.html.

“Cyber-Physical Systems and Autonomy - Highlighting Raytheon Company’s Work.” 2017. Business-Higher Education Forum. http://www.bhef.com/sites/default/files/BHEF_17_Case%20%20Study_Raytheon.pdf.

“DARPA CGC ~ Shellphish.” n.d. Accessed March 21, 2019. http://shellphish.net/cgc/.

Davidson, Jack W. 2017. “DARPA Cyber Grand Challenge.” 2017. https://www.cs.virginia.edu/~jwd/page/page-5/.

Fazeldehkordi, E., O. A. Akanbi, and Iraj Sadegh Amiri. 2014. *A Machine-Learning Approach to Phishing Detection and Defense*. Syngress.

Fraze, Dustin. n.d. “Cyber Grand Challenge (CGC).” Cyber Grand Challenge (CGC) (Archived). Accessed March 21, 2019. https://www.darpa.mil/program/cyber-grand-challenge.

Gil, Laurent, and Allan Liska. 2019. *Security with AI and Machine Learning*. O’Reilly Media, Inc.

“GrammaTech.” 2019. *Wikipedia*.
https://en.wikipedia.org/w/index.php?title=GrammaTech&oldid=887172690 .

“GrammaTech’s Team TECHx Places Second in DARPA’s Cyber Grand Challenge.” Accessed March 21, 2019. http://news.grammatech.com/grammatech-team-techx-places-second-in-darpa-cyber-grand-challenge.

Graves, Alex, Greg Wayne, and Ivo Danihelka. 2014. “Neural Turning Machines.” Google DeepMind, London, UK.
https://arxiv.org/pdf/1410.5401.pdf%20(http://Neural%20Turning%20Machines)%20.

Greengard, Samuel. 2016. “Cybersecurity Gets Smart.” *Communications of the ACM* 59 (5): 29–31. https://doi.org/10.1145/2898969.

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. “Long Short-Term Memory.” *Neural Computation* 9 (8): 1735–80.

“HP 2018 Cybersecurity Guide: Hackers and Defenders Harness Design and Machine Learning.” 2018. HP Development Company, L.P.

Hutson, Matthew. 2018. “Hackers Easily Fool Artificial Intelligences.” *Science* 361 (6399): 215–215. https://doi.org/10.1126/science.361.6399.215.

“IBM X-Force Threat Intelligence Index 2019.” 2019. Armonk, NY: IBM Security.
https://www.ibm.com/downloads/cas/ZGB3ERYD.

Introduction to Artificial Intelligence for Security Professionals. 2017. Irvine, CA: The Cylance Press.

Katzir, Ziv, and Yuval Elovici. 2018. “Quantifying the Resilience of Machine Learning Classifiers Used for Cyber Security.” *Expert Systems with Applications* 92 (February): 419–29. https://doi.org/10.1016/j.eswa.2017.09.053.

Kissel, Richard, Kevin Stine, Matthew Scholl, Hart Rossman, Jim Fahlsing, and Jessica Gulick. 2008. “Security Considerations in the System Development Life Cycle.” NIST Special Publication (SP) 800-64 Rev. 2. National Institute of Standards and Technology.
https://doi.org/10.6028/NIST.SP.800-64r2.

Klarreich, Erica. 2016. “Learning Securely.” *Communications of the ACM* 59 (11): 12–14. https://doi.org/10.1145/2994577.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. “Deep Learning.” *Nature* 521 (7553): 436–44. https://doi.org/10.1038/nature14539.

Liu, Xiaoxiao, Qingyang Xu, and Ning Wang. 2018. “A Survey on Deep Neural Network-Based Image Captioning.” *The Visual Computer*, June. https://doi.org/10.1007/s00371-018-1566-y.

“Malwr Alternatives and Similar Websites and Apps - AlternativeTo.Net.” n.d. AlternativeTo. Accessed April 9, 2019. https://alternativeto.net/software/malwr/.

McCarthy, John, Marvin L. Minsky, Nathaniel Rochester, and Claude E. Shannon, “A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence. August 31, 1955.” 2006. AI Magazine 27 (4).

McCulloch, Warren S, and Walter Pitts. 1990. “A Logical Calculus of the Ideas Immanent in Nervous Activity.” *Bulletin of Mathematical Biology* 52 (1/2): 99–115.

Mühlenbein, Heinz. 2009. “Computational Intelligence : The Legacy of Alan Turing And.” In *Computational Intelligence, Intelligent Systems Reference Library*, edited by C. L. Mumford and L. C. Jain, 1:23–43. Springer Berlin Heidelberg.

“Oracle and KPMG Cloud Threat Report 2019.” 2019. Oracle.

Papernot, Nicolas, Patrick McDaniel, and Ian Goodfellow. 2016. “Transferability in Machine Learning: From Phenomena to Black-Box Attacks Using Adversarial Samples.” *ArXiv:1605.07277 [Cs]*, May. http://arxiv.org/abs/1605.07277.

“Perspectives on Research in Artificial Intelligence and Artificial General Intelligence Relevant to DoD.” 2017. JSR-16-Task-003. McLean, Virginia: The MITRE Corporation, JASON Program Office.

Pino, Robinson E., Alexander Kott, and Michael Shevenell, eds. 2014. *Cybersecurity Systems for Human Cognition Augmentation*. Advances in Information Security 61. New York: Springer.

“Raytheon: The Bot Defenders - Humans and Machines Team up to Defeat Cyber Attacks.” n.d. Accessed March 21, 2019. https://www.raytheon.com/cyber/news/feature/cyber_human_machine.

Scarfone, K A, and P M Mell. 2007. “Guide to Intrusion Detection and Prevention Systems (IDPS).” NIST SP 800-94. Gaithersburg, MD: National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-94.

Scarfone, Karen, and Peter Mell. 2012. “Guide to Intrusion Detection and Prevention Systems (IDPS).” NIST Special Publication (SP) 800-94 Rev. 1 (Draft). National Institute of Standards and Technology. https://csrc.nist.gov/publications/detail/sp/800-94/rev-1/draft.

Schuman, Catherine D., Thomas E. Potok, Robert M. Patton, J. Douglas Birdwell, Mark E. Dean, Garrett S. Rose, and James S. Plank. 2017. “A Survey of Neuromorphic Computing and Neural Networks in Hardware.” *ArXiv:1705.06963 [Cs]*, May. http://arxiv.org/abs/1705.06963.

Shoshitaishvili, Yan, Antonio Bianchi, Kevin Borgolte, Amat Cama, Jacopo Corbetta, Francesco Disperati, Audrey Dutcher, et al. 2018. “Mechanical Phish: Resilient Autonomous Hacking.” *IEEE Security & Privacy* 16 (2): 12–22. https://doi.org/10.1109/MSP.2018.1870858.

Stephens, Nick, John Grosen, Christopher Salls, Andrew Dutcher, Ruoyu Wang, Jacopo Corbetta, Yan Shoshitaishvili, Christopher Kruegel, and Giovanni Vigna. 2016. “Driller: Augmenting Fuzzing Through Selective Symbolic Execution.” In *Proceedings 2016 Network and Distributed System Security Symposium*. San Diego, CA: Internet Society. https://doi.org/10.14722/ndss.2016.23368.

Shoshitaishvili, Yan, Ruoyu Wang, Christopher Salls, Nick Stephens, Mario Polino, Audrey Dutcher, John Grosen, et al. 2016. “SOK: (State of) The Art of War: Offensive Techniques in Binary Analysis.” In *Proceedings of the IEEE Symposium on Security and Privacy*, 1:138–57. https://doi.org/10.1109/SP.2016.17.

Song, J., and J. Alves-Foss. 2015. “The DARPA Cyber Grand Challenge: A Competitor’s Perspective.” *IEEE Security Privacy* 13 (6): 72–76. https://doi.org/10.1109/MSP.2015.132.

Song, J., and J. Alves-Foss. 2016. “The DARPA Cyber Grand Challenge: A Competitor’s Perspective, Part 2.” *IEEE Security Privacy* 14 (1): 76–81. https://doi.org/10.1109/MSP.2016.14.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting.” *Journal of Machine Learning Research* 15 (6): 1929–58.

Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. 2014. “Sequence to Sequence Learning with Neural Networks.” *ArXiv:1409.3215 [Cs]*, September. http://arxiv.org/abs/1409.3215.

Taleqani, A. R., K. E. Nygard, R. Bridgelall, and J. Hough. 2018. “Machine Learning Approach to Cyber Security in Aviation.” In *2018 IEEE International Conference on Electro/Information Technology (EIT)*, 0147–52. https://doi.org/10.1109/EIT.2018.8500165.

Tamburello, Paul, and Peter Guerra. 2018. *Modernizing Cybersecurity Operations with Machine Intelligence*. O’Reilly Media, Inc.

“Tesla V100 Application Performance Guide: Deep Learning and HPC Applications.” 2018. NVIDIA. https://images.nvidia.com/content/pdf/v100-application-performance-guide.pdf.

Tucker, Patrick. 2019. “You Have 19 Minutes to React If the Russians Hack Your Network.” Defense One. February 19, 2019. https://www.defenseone.com/technology/2019/02/russian-hackers-work-several-times-faster-chinese-counterparts-new-data-shows/154952/.

Turing, A. M. 2009. “Computing Machinery and Intelligence.” In *Parsing the Turing Test*, edited by R. Epstein, G. Roberts, and G. Berger. Springer, Dordrecht. https://link-springer-com.proxy.lib.utk.edu/content/pdf/10.1007%2F978-1-4020-6710-5_3.pdf.

“Vetting Commodity IT Software and Firmware (VET).” n.d. Accessed March 25, 2019. https://www.darpa.mil/program/vetting-commodity-it-software-and-firmware.

Vinyals, Oriol, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2014. “Show and Tell: A Neural Image Caption Generator.” *ArXiv:1411.4555 [Cs]*, November. http://arxiv.org/abs/1411.4555.

“VirusTotal.” n.d. Accessed April 9, 2019. https://support.virustotal.com/hc/en-us.

Wang, Ruoyu, Yan Shoshitaishvili, Antonio Bianchi, Aravind Machiry, John Grosen, Paul Grosen, Christopher Kruegel, and Giovanni Vigna. 2017. “Ramblr: Making Reassembly Great Again.” In *Proceedings 2017 Network and Distributed System Security Symposium*. San Diego, CA: Internet Society. https://doi.org/10.14722/ndss.2017.23225.

Weston, Jason, Sumit Chopra, and Antoine Bordes. 2015. “Memory Networks.” Facebook AI Research, New York, NY. https://arxiv.org/pdf/1410.3916.pdf.

Whittaker, Colin, Brian Ryner, and Marria Nazif. 2010. “Large-Scale Automatic Classification of Phishing Pages.” *NDSS ’10*, 14.

Wilkins, Jonathan. 2018. “Is Artificial Intelligence a Help or Hindrance?” *Network Security* 2018 (5): 18–19. https://doi.org/10.1016/S1353-4858(18)30046-1.

Xin, Y., L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang. 2018. “Machine Learning and Deep Learning Methods for Cybersecurity.” *IEEE Access* 6: 35365–81. https://doi.org/10.1109/ACCESS.2018.2836950.

Xu, Kelvin, Jimmy Lei, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio. n.d. “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention.” *Proceedings of the 32nd International Conference on Machine Learning*, JMLR: W&CP, 37: 10.

Yampolskiy, Roman V., and M. S. Spellchecker. 2016. “Artificial Intelligence Safety and Cybersecurity: A Timeline of AI Failures.” *ArXiv:1610.07997 [Cs]*, October. http://arxiv.org/abs/1610.07997.

Yuan, X., P. He, Q. Zhu, and X. Li. 2019. “Adversarial Examples: Attacks and Defenses for Deep Learning.” *IEEE Transactions on Neural Networks and Learning Systems*, 1–20. https://doi.org/10.1109/TNNLS.2018.2886017.

Zalewski, Michal. n.d. “American Fuzzy Lop.” Accessed March 24, 2019. http://lcamtuf.coredump.cx/afl/.

Appendix A. Comments on "Deep Learning"

The following are brief comments on an influential paper by three of the primary architects of deep learning (LeCun, Bengio, and Hinton 2015). The focus of these comments is on the potential of deep learning approaches in methods to automatically comment source code or, put another way, to translate computer source code into descriptive text.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. "Deep Learning." *Nature* 521 (7553): 436–44.

This paper provides a survey and overview of deep learning research results and applications up to about 2015. The authors are well-known researchers[21] in the deep learning community representing Facebook and NYU (LeCun), Université de Montréal (Bengio), and Google and University of Toronto (Hinton). The paper has significant value, though the authors should not be considered completely objective. The bibliography, in particular, is quite good, although the authors extensively quote their own work.

Although the source data are different, several of the topics and research results discussed in this paper appear to be relevant to analysis tools such as vulnerability assessment and the possibility of automatic comment generation from source code. Note the following in particular:

1. Convolutional neural networks and pooling (aggregation) can localize features in input examples (multiple locations in the paper).
2. Very promising results have been published on the automatic captioning of images using deep learning structures with convolutional neural networks (p. 440, Figure 3).
3. Specific features of neural network architectures and training methods that have improved learning rates and accuracy are listed, including:
    a. The use of the half-wave rectifier (ReLU) function speeds up training using back propagation (p. 438).
    b. The use of stochastic gradient methods is prevalent (p. 438).
    c. Gradient descent (back propagation) methods tend to be effective because of the prevalence of more or less equivalent local minima in the cost function (p. 438).

[21] The authors received the Turing award in 2019 for their work in deep learning. See: "A.M. Turing Award." n.d. Accessed April 9, 2019. https://amturing.acm.org/.

d. The use of dropout methods has been beneficial to keep training procedures from over-fitting the data (p. 440).

e. The combination of LSTM with recursive neural networks facilitates learning of sequence information; this is of primary interest in natural language and source code processing (p. 442).

4. The availability and applicability (via CUDA and OpenCL) of graphic processing units (GPUs) prompted a revival in deep learning research and development activities (p. 439) This appendix provides some background, which indicates a roughly order of magnitude speed-up in computations (measured by floating point operations per second, or FLOPS) using GPUs.

5. A discussion is provided of neural networks' ability to learn similarities among words in a dictionary of words, and emerging technologies' applicability to natural language processing, including the Neural Turing Machine and Memory Networks is discussed.

This being said, the authors perhaps over-sell the potential of deep learning methods while downplaying their disadvantages. Primary among the disadvantages are their complexity (measured by both the number of layers in these networks and the number of network parameters that must be trained), the need to hand-design the overall structure of deep learning networks (the functions and types of the layers), and the black-box nature of computational structures built using deep learning (difficulty of explaining why a specific result is reached). However, modern computer hardware and GPUs in particular have made deep learning technologies feasible, and emerging toolsets (such as TensorFlow, Keras, and PyTorch) and cloud service providers (such as Google Cloud, Amazon Web Services, and Microsoft Azure) have made their application accessible to a wide audience of developers. Although cost can still be an issue, the cost of entry is much lower than it was only a few years ago.

Deep learning refers to computational structures that incorporate multiple layers of neural networks that can be trained using back propagation (a gradient-based optimization method that operates on the weights of the neural network(s)). Each layer can have a different purpose, and alternating structures of RNNs, CNNs, and aggregators (such as max pooling, or computation of a maximum over a set of outputs from a previous neural network layer to reduce dimensionality) are often used. The combination of a CNN and an aggregation operation (referred to as pooling in the paper) is called a ConvNet in the paper.

An RNN incorporates memory whose values are fed back to the inputs of the network and updated by the network's outputs. A CNN is a type of finite impulse response filter that convolves a finite set of weights (in 1, 2, or 3 dimensions) with the inputs and combines with a nonlinear output function to produce the CNN's output.

Deep learning usually utilizes supervised learning, but can perform unsupervised learning, for example by optimizing over the weights of a CNN in a ConvNet – possibly with some assistance from other algorithms such as clustering (e.g., K-means). Deep learning structures have proved effective in diverse areas including speech processing, image recognition, handwritten digit classification, natural language processing, and translation.

In addition to the extensive bibliography in this paper, there are several takeaways that should be of interest. These are listed below, with short comments, in no particular order.

1. Neural networks are typically composed of elements that compute a weighted sum of selected inputs followed by a limiter function that limits the elements' outputs to a finite range. Most research over the first few decades of interest in neural networks utilized a sigmoid, or S-shaped, nonlinearity for the limiter. Part of the reasoning for this was its differentiability and the fact that gradients of the output values provided information to influence the directions of changes for all of the network's weights during back propagation. The authors note that "[a]t present, the most popular non-linear function is the rectified linear unit (ReLU), which is simply the half-wave rectifier f(z)=max(z,0)" (p. 438). When the weighted sum is non-negative, the gradient provides no information (because the derivative is zero), which effectively turns off changes to a subset of weights, so this observation is somewhat surprising. However, the rectifier also probably serves to "sparsify" the network's weights, at least with respect to back propagation, and that may lead to the improved training characteristics the authors cite: "…the ReLU typically learns much faster in networks with many layers, allowing training of a deep supervised network without unsupervised pre-training" (p. 438).

2. The authors note (correctly) that "… it was commonly thought that simple gradient descent would get trapped in poor local minima – weight configurations for which no small change would reduce the average error[,]" but that "[i]n practice poor local minima are rarely a problem with large networks" (p. 438). Back propagation is a first order optimization algorithm (depending only on the gradient, or first derivatives), which makes it both robust (reliably converging to a local optimum) and slow (with long training times). It is also a local optimization algorithm, as opposed to stochastic optimization methods such as simulated annealing and genetic programming, which explore much more diverse areas of the parameter space. Birdwell et al. (1990, 1992) found that overparameterization could introduce a large number of more or less equivalent local minima, as the authors state, so the observation is not surprising, although it is still possible for any local optimization algorithm to converge to a poor local minimum.

3. The authors note that "most practitioners use a procedure called stochastic gradient descent (SGD)" (p. 437) to train networks from examples. As described, this is the repetitive random selection of small sets of examples from the training set, followed by

computation of gradients and update of weights via back propagation using only each small set, and resulting in a decision whether to terminate training with a solution. In traditional optimization methods (examples include linear programming, nonlinear programming, and convex programming), algorithms are used to select a direction of descent or ascent (for a minimization program or a maximization problem, respectively) without explicitly computing a gradient or Hessian (matrix of second order derivatives).[22] As described in this article, random selection of examples is one approach to use when computing the gradient using a training set. Random selection does not guarantee descent, which is always desirable; however, on average, the SGD method does move weights toward a lower cost (or higher reward) value of the objective function.

4. One criticism of deep learning methods is that they are primarily utilized for supervised learning (using labeled or classified training data). The authors point to work from 2005–2006 where unsupervised learning was performed using multi-layer networks and "pre-training." Although this is interesting, it should also be noted that the authors are quoting their own work.

5. The authors provide a good discussion of CNNs and the combination of CNNs and aggregation (pooling) used to identify features for further processing and classification (p. 439). They provide references to the similarities between these structures and the structures of the visual cortex from the neuroscience literature. Finally, they provide references to early applications of CNNs in speech recognition (starting in the 1990s), in document reading (including handwriting recognition), and in object detection and image recognition.

6. The paper includes a section on the use of deep ConvNets for image understanding, with a primary application to vision systems for transportation. The authors state that "ConvNets were largely forsaken by the mainstream computer-vision and machine-learning communities until the ImageNet competition in 2012. When deep convolutional networks were applied to a data set of about a million images from the web that contained 1,000 different classes, they achieved spectacular results, almost halving the error rates of the best competing approaches" (p. 440). The authors claim the success depended upon the "efficient use of GPUs [graphics processing units], ReLUs, a new regularization technique called dropout, and techniques to generate more training examples by deforming the existing ones", and that "ConvNets are now the dominant approach for almost all recognition and detection tasks" (p. 440).

Figure 3 on p. 440 is relevant to source code comment generation. The images in the figure have been automatically captioned using a ConvNet approach. Two references in

[22] See, for example, (Bertsekas 1999).

the bibliography (Xu et al.[23] and Vinyals et al.[24]) refer to caption generation in their titles. Figure 3 is a copy of Figure 4 from another paper found on the Web, which is related to these papers but has a slightly different title.[25] Note that an author, Bengio, is also an author of all of these papers. A library search located a 2018 journal paper surveying recent results in automatic caption generation.[26]

GPUs and ReLUs are discussed elsewhere in these notes. The dropout method (p. 440) was developed by Srivastava et al. (including G. Hinton), and, according to the referenced paper,[27] the key idea "is to randomly drop units (along with their connections) from the neural network during training. This prevents units from co-adapting too much" (p. 1929). The deformation of existing examples in a training set to create new examples, and other methods such as random modifications and cross-overs, are now widely known.

A final note of use from this section relates to the size and complexity of ConvNet architectures in use by the date of the paper: "Recent ConvNet architectures have 10 to 20 layers of ReLUs, hundreds of millions of weights, and billions of connections between units. Whereas training such large networks could have taken weeks only two years ago, progress in hardware, software and algorithm parallelization have reduced training times to a few hours" (p. 440). Hardware issues are discussed below. Current algorithm parallelization methods now play a large role in the application of deep learning networks, including TensorFlow[28], PyTorch[29], and Keras[30]. (There are others; see https://en.wikipedia.org/wiki/Comparison_of_deep-learning_software for a listing and comparison of features. However, note that this Wikipedia page also lists software packages that are no longer maintained, so caution is warranted.)

7. The article implies that deep learning approaches might have died out (p. 439) because of the complexity of their training algorithms (due to their extraordinarily large parameter spaces) if not for the discovery that these algorithms could be implemented using graphic processor units (GPUs), the rapid evolution of GPU hardware, and the use of parallel computer architectures. The following information is from a variety of sources, provided to supplement the perspectives of the article's authors.

---

[23] Xu, K. et al. Show, attend and tell: Neural image caption generation with visual attention. In *Proc. International Conference on Learning Representations* http://arxiv.org/abs/1502.03044 (2015). [reference 86 in the paper]

[24] See (Vinyals, Toshev, Bengio, and Erhan 2014). [reference 102 in the paper]

[25] See (Xu, Lei, Kiros, Cho, Courville, Salakhutdinov, Zemel, and Bengio n.d.).

[26] See (Liu, Xu, and Wang 2018).

[27] See (Srivastava, Hinton, Krizhevsky, Sutskever, and Salakhutdinov 2014).

[28] See https://www.tensorflow.org/.

[29] See https://pytorch.org/.

[30] See https://keras.io/.

The present generation of GPUs offer roughly one order of magnitude improved performance (decreased training time) over CPUs, but there are many variables that affect performance. GPUs operate in combination with at least one CPU (and possibly many CPU cores). A typical configuration combines one or two GPUs (often as PCIe daughterboards to the computer's motherboard) with a conventional (CPU-based) computer having one to four CPUs, each having 2-32 (or possibly more) cores. There are many opportunities for bandwidth limitations to create communications bottlenecks between the CPUs, GPUs, memories, and input/output devices, and these bottlenecks can limit performance. Multiple copies of this configuration can be grouped into a parallel computer using a high-bandwidth interconnect such as the various types of Infiniband or Ethernet, scaling to the (currently) fastest computers in the world. A researcher or developer using deep learning technologies has many choices including either GPU-enabled technologies or more conventional computer architectures. However, current "conventional wisdom" appears to favor using computers with GPUs to train and test deep learning algorithms.

GPUs currently enable peak performance per GPU of roughly 8 teraflops (1 TFLOPS = 10^{12} floating point operations per second, or FLOPS),[31,32] and, when combined with parallel processing methods, peak performance in excess of a hundred petaflops (1 PFLOPS = 10^{15} FLOPS).[33] By way of comparison, the dual Xeon 2690v4 Intel Core i9 processor with 28 cores (not virtualized) has been benchmarked at 1.123 TFLOPS.[34] More traditional (affordable) computer processor units (CPUs) provide a few to several hundred gigaflops (1 GFLOPS = 10^9 FLOPS) per CPU, scaling to tens of TFLOPS to (perhaps) the low PFLOPS in large parallel computers.

It is possible to utilize cloud resources to develop deep learning models and applications, and these resources are likely to be utilized by most members of the deep learning research and development community. For example, Amazon Web Services (AWS) offers the PC2 P3 compute instances, each with up to 8 NVIDIA V100 GPUs,[35]

---

[31] All floating point performance figures are for double precision computations. Single precision performance will be higher. Note that computational performance for computers (including those with attached GPUs) and CPUs are measured using the high-performance Linpack benchmark, while computational performance for stand-along GPUs are theoretical maximum values provided by the manufacturers that are likely not fully realized using the Linpack benchmark.

[32] The NVIDIA Tesla V100 GPU has a claimed performance of 7.8 TFLOPS. See ("Tesla V100 Application Performance Guide: Deep Learning and HPC Applications" 2018).

[33] The fastest computer in the world at present is Summit at Oak Ridge National Laboratory (USA), with a peak performance of 143.5 PFLOPS on the high-performance Linpack benchmark using 2.4 million cores and NVIDIA Volta GV100 GPUs, according to the November, 2018 TOP500 list (https://www.top500.org/lists/2018/11/).

[34] See https://www.pugetsystems.com/labs/hpc/Intel-Core-i9-7900X-and-7980XE-Skylake-X-Linux-Linpack-Performance-1059/.

[35] See https://aws.amazon.com/ec2/instance-types/p3/.

providing a theoretical peak performance of over 60 TFLOPS. Google Cloud[36] and Microsoft Azure[37] provide similar capabilities.

8. The last substantive section of the paper discusses applications of deep learning to language processing. The first example the paper discusses work by one of the authors (Bengio) that uses a multi-layer neural network to predict the next word in a sequence of words. Words are represented as binary vectors (each element of which corresponds to a word in a dictionary) with a single non-zero element (that is unity) corresponding to each word. A note in the paper's bibliography states that "[t]his paper introduced neural language models, which learn to convert a word symbol into a word vector or word embedding composed of learned semantic features in order to predict the next word in a sequence" (p. 443). The authors are somewhat disingenuous because binary encoding of words in sentences (along with the use of principal component analysis, or PCA), was utilized at least as far back as the 1980s with the introduction of latent semantic analysis (LSA) for text search. Admittedly, however, the objective at that time was search and retrieval rather than next word prediction. What is interesting, however, in the authors' discussion is the use of vectors representing probabilities (or, more accurately, estimated or predicted frequencies of occurrence) of dictionary words to discover associations or patterns among the words. See, for example, Figure 4 of the paper. (p. 441)

This section of the paper continues with a discussion of RNNs and training of such networks and the introduction of LSTM networks to facilitate training for remembering information over a long time delay[38]. The authors state: "LSTM networks have subsequently proved to be more effective than conventional RNNs, especially when they have several layers for each time step, enabling an entire speech recognition system that goes all the way from acoustics to the sequence of characters in the transcription. LSTM networks or related forms of gated units are also currently used for the encoder and decoder networks that perform so well at machine translation" (p. 442). The authors reference, among others, Sutskever's paper on sequence to sequence learning.[39]

Finally, the authors discuss recently developed methods to incorporate memory in RNNs, including a Neural Turing Machine (NTM)[40] and Memory Networks.[41] An NTM was shown capable of learning and performing an algorithm (sorting numbers), and a Memory Network was able to correctly answer the question "Where is Frodo now?" after

[36] See https://cloud.google.com/products/ai/.

[37] See https://azure.microsoft.com/en-us/overview/ai-platform/.

[38] See (Hochreiter and Schmidhuber. 1997).

[39] See (Sutskever, Vinyals, and Le 2014).

[40] See (Graves, Wayne, and Danihelka 2014).

[41] See (Weston, Chopra, and Bordes 2015).

being trained on a 15-sentence version of *The Lord of the Rings*. These approaches appear to show promise, but further investigation is necessary to assess their potential.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE (DD-MM-YY)	2. REPORT TYPE	3. DATES COVERED (From – To)
00-06-19	Non-Standard	

4. TITLE AND SUBTITLE
Utility of Artificial Intelligence and Machine Learning in Cybersecurity

5a. CONTRACT NUMBER
HQ0034-14-D-0001

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBERS

6. AUTHOR(S)
John D. Birdwell, George L. Kennedy, Francisco L. Loaiza, Dale Visser

5d. PROJECT NUMBER
DI-5-4630

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESSES
Institute for Defense Analyses
4850 Mark Center Drive
Alexandria, VA 22311-1882

8. PERFORMING ORGANIZATION REPORT NUMBER
NS D-10694

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Garnard W. Burnside II, Project Director, Enterprise Services
Program Executive Office Enterprise Information Systems
10119 Beach Road
BLDG 322, Room 2212
Fort Belvoir, VA 22060-5801

10. SPONSOR'S / MONITOR'S ACRONYM
PEO EIS

11. SPONSOR'S / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
Project Leader: Francisco L. Loaiza

14. ABSTRACT
This paper discusses selected publications that relate cybersecurity to artificial intelligence (AI) in general and to machine learning (ML) specifically. The focus is cybersecurity in the context of software development and lifecycle environments (SDLE) and their products. Because of the large volume of publications in this area, the survey includes publications that are themselves surveys of specialized subjects. A few papers are cautionary, pointing out that systems trained using AI/ML can be fooled; one of these papers investigates methods for the design of classifiers that exhibit resilience to adversarial actions and points to other literature in this emerging field. Several references are books, and the discussions of these provide some guidance for those who might be interested in further reading across the breadth of the field. A few relevant publications from the NIST 800 series are included as background to provide an appropriate setting for the discussion of AI/ML as applied to cybersecurity.

15. SUBJECT TERMS
artificial intelligence (AI), machine learning (ML), cybersecurity, software development and lifecycle environments (SDLE)

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE	Unlimited	59	Garnard W. Burnside II, Project Director, Enterprise Services
Unclassified	Unclassified	Unclassified			19b. TELEPHONE NUMBER (Include Area Code) 703-704-3716