MixMode™

SELF-SUPERVISED LEARNING – AI FOR COMPLEX NETWORK SECURITY

WHITEPAPER

PETER R. STEPHENSON, PHD
UNIVERSITY OF LEICESTER
LEICESTER, UK

ABSTRACT

Artificial Intelligence – or AI – has become a buzzword since it emerged in the 1950s. However, all AI systems are not created equal. In this white paper Dr. Peter Stephenson explains the different "waves" of artificial intelligence[1]. He uses the DARPA definitions for each of these waves[2]. In this paper we will continue that discussion by adding our own context to the problem of what constitutes useful AI as it relates to cyber security.

Of necessity we must move beyond simple definitions to the underlying contexts that surround them. In this paper we will look at those contexts, apply them to the field of cyber security in general and network monitoring and intelligent interpretation – particularly the ability of a monitoring systems to produce actionable intelligence – in particular, and demonstrate our findings when analyzing MixMode.

ABOUT DR. STEPHENSON

Peter Stephenson received his PhD in computing for his work in investigating digital incidents in large scale computing environments at Oxford Brookes University in the UK. He holds a master's degree in diplomacy with a concentration in terrorism, cum laude, from Norwich University in Vermont and has been practicing in the security, forensics and digital investigation fields for over 55 years. He was technical editor for SC Magazine and supervised or performed security product testing and reviewing for over a decade averaging in the hundreds of products seen per year. He currently is working towards a second PhD, this time in law focusing on jurisdiction in cyberspace. He has authored or contributed to 20 books and hundreds of articles, many for peer reviewed journals.

[1] "MixMode Artificial Intelligence for Cyber Security"
[2] Wave 1 – Expert knowledge rules & thresholds, Wave 2 – Statistical learning training data and Wave 3 – Contextual adaptation context-aware

SELF-SUPERVISED LEARNING - AI FOR COMPLEX NETWORK SECURITY

I. COMPLEXITY, EMERGENCE AND ENTROPY – CONCEPTS THAT INFORM OUR VIEW OF CYBER SCIENCE

These terms can become very complex for students of mathematics, the social sciences and physics (among other disciplines) but for our purposes we are concerned with rather small subsets of each field. Each of these terms helps us understand why a simple approach – e.g., current generation or Waves 1 and 2 AI – simply won't work in today's and tomorrow's world.

We begin by understanding the environment in which all modern computing systems exist. Robert Rosen defined a complex system as one that could not be simulated or modeled[3]. Dr. Rosen worked in the biological sciences and he made a clear distinction between complex and simple.

In the predictable world of Wave 1 and Wave 2, everything can be modeled. It can be modeled because it is programmatic. If data were generated by a model/program it can itself be modeled. In the Wave 3 world we may assume that there will be random events. Random events, by definition, cannot necessarily be modeled.

First, where do these random events come from? Imagine a next generation bot, programmed by its maker to achieve a mission – exfiltrate a certain type of data, for example. The bot is programmed with next generation – Wave 3 – AI such that it enters the target network, explores the network and, gaining the context that impacts its mission learns what it must learn to achieve the mission.

Our bot has begun to "emerge". Emergence occurs when an entity enters another entity and becomes integral with it. Emergence – or an emergent property – simply is a case where something is introduced into a system for the first time and it evolves to become part of the system[4]. Our bot is doing exactly that.

Let's take the analogy a step further and imagine what our bot's mission is.

Our bot is going to engage in phishing, specifically, lateral phishing. Lateral phishing involves staying within the network and using data from network users to phish other network users. An example might be an email from the CEO to the controller ordering the payment of a large consulting contract. The contract, however, does not exist and the consultant is a hacker outside of the network who expects to receive the large payment. What must our emergent bot do to accomplish this?

[3] ROBERT ROSEN, ESSAYS ON LIFE ITSELF (Columbia University Press 1 ed. 2000).
[4] Peter Stephenson, International Private Law as a Model for Private Law Jurisdiction in Cyberspace (Legal Issues Journal, Volume 7, Issue 2 July 2019)

First, our bot must learn the email system including addressing norms and who in the organization is responsible for what. Second, it must learn the way certain employees communicate, what phrases they use, how they address other employees, etc. Once it knows all of this it can complete its mission. Perhaps part of its mission is to destroy itself once the email has been sent and acknowledged.

All of this is complex, emergent behavior. The bot has become another user in the system behaving just as a user would and spoofing whomever and whatever it needs to do to accomplish its mission. It has no external instructions, no bot herder to tell it what to do and no command and control server to report to for instructions.

In 1948, Charles Shannon posed a theory of communication subsequently known as the Shannon Information Theory. Part of Shannon's theory is the concept of entropy. While entropy can become quite complicated, for our purposes it is enough to say that entropy is a measure of randomness. The more random something is the more difficult it is to model. The more difficult it is to model, the closer it comes to being complex.

If we could predict our bot's behavior, we could write a Wave 1 or Wave 2 program that would make observations about the bot and attempt to identify and stop it. The problem is that randomness is not, by definition, predictable. Wave 1 or 2 defenses might get it right or they might not.

The more random the bot's behavior becomes (the higher its entropy is) as it emerges into the infrastructure and carries out its mission the more complex its behavior also becomes. Shannon's model comprises a sender of information and a receiver. In the middle it includes a noise source. The noise source tends to confound the clear reception of the information from the sender. People inject noise into a network.

For example, our CEO may send out Christmas greetings using entirely different terms and phrases than she would when requesting the consulting payment to be made. Our bot must learn how, when and what to inject to create that noise.

The point? If the bot achieves true Rosennean complexity it will be nearly impossible for a Wave 1 or 2 AI to catch it reliably. Only a defense that is itself emergent will succeed. In order for that to occur the defense must learn from its environment in the same way that the adversary's bot does.

II. MACHINE LEARNING, DEEP LEARNING AND NEURAL NETWORKS, OH MY!

There are three more terms that get bandied about without much real definition in typical marketing materials. They, like AI, have become buzzwords without usually giving a lay description of what they really mean (or don't mean). Since Russell's white paper gives us a great start, let's take a little deeper dive. This is important because, as we progress to the all-important "what's in it for me?" question, we need a little technical background to help us get past the jargon.

If we think of machine learning (ML) as fact gathering and Deep Learning (DL) as interpretation we'll be pretty close on the meat of the terms. It's really a bit more than that, of course.

Machine learning does, in fact, gather data but it does not require explicit programming to do so. For example, first generation anti-virus products worked on a basis of signatures. They looked for a signature – a specific bit pattern – to identify a virus.

That was pretty simple to fool and the moment a new .dat file emerged from a vendor, a set of changes to the viruses it could identify appeared in the criminal hacker underground. The virus writers made the changes and the AV product's efficacy was diminished substantially until the next round at which time, of course, the cycle repeated. These .dat files and their countermeasures by the adversary were not ML. They were, simply, pattern recognition.

The next step the AV vendors took was the introduction of heuristics and behavior-based analysis. These at their inception were crude examples of ML. The AV was observing known signatures and looking for things that looked or behaved a bit like them. So the fact gathering was there but there was a bit more intelligence applied to it.

Deep learning makes decisions based upon the data it sees and the data that it doesn't see but infers from what it does see. This became useful in the AV industry when the adversary introduced polymorphic viruses. These are viruses that change their appearance on the fly and not always in the same way.

For example, a polymorph may be encrypted to hide its signature. At some point in its execution cycle it decrypts itself, does its damage and re-encrypts, this time using a different key. An ML system would have a hard time with this because of the seemingly random changes but a DL system might look at the code, decide that it's encrypted and make some decisions as to what to do about it.

Some early systems actually put the code in a sandbox and single-stepped through it until decryption occurred and then tried to identify the malware. This was inefficient, slowed down or crashed the host application and produced false positives, but the landscape was changing. Wave 3 approaches address the same problem far more efficiently and without false returns.

Neural networks (NN) simply are an extension of machine learning where the AI system uses the data collected by the ML and attempts to analyze it in much the same way a human would.

The NN, then, tries to extract knowledge from all of the data collected and inferred by the AI and make decisions based upon those data. What it does not do is take the context of the data collected and inferred in order to create novel decisions that, in themselves, may predict behavior.

III. LEARNING AND TRAINING – THE CORE ISSUE IN WAVE 3 AI

Now we get to the meat of the subject… how an AI system gathers, infers and uses its data. We have discussed the gathering of the data, but we have yet to address how it is used. More important, we have yet to discuss how an AI system ingests those data. That is called, not surprisingly, learning. And, an AI system learns by being trained. How it is trained is very important because one approach mires us in the pre-Wave 3 world while the other releases the magic of the third wave self-supervised AI.

Training consists of two types: supervised and self-supervised. The difference between the two is as simple as the terminology implies.

Imagine your high school years. You walk into a classroom on the first day of classes, your teacher gives you a text book and a reading assignment. She then gives you a lecture on the topic and sends you off. At some point you'll take a test over what you've learned.

An excellent example – pre-high school to be sure – was the old way of learning the multiplication tables: constant repetition and rote memorization. That is supervised training.

Self-supervised training, on the other hand, is achieved through observation. We observe and draw conclusions from what we observe. The more we observe, the more our conclusions are refined.

For example, as small children we are introduced to unicorns. They are in our fairy tales and Mom and Dad may make up stories about the to help us

wind down our day and go to sleep. When we learn to read we see stories about unicorns and learn about where they live, what they eat and what kind of magic they possess. But we are a little older and we're a bit skeptical because we haven't actually seen a unicorn yet.

Time passes and we see movies with unicorns but we observe that they always are cartoons. We go to the zoo and there are no unicorns. Our perception of unicorns is being shaped by our environment and, eventually, based upon contextual observation, we conclude that there probably are no unicorns. That is self-supervised training. We learn, not from a structured training set but from our environmental observation.

IV. WHY TRAINING MATTERS – AND HOW THE ADVERSARY TAKES ADVANTAGE OF IT

There are over 1,200 peer-reviewed papers written on the subject of adversarial AI[5] going back to 2014. That means that there are many tested (mathematically and in the lab, usually) attacks that can, potentially, succeed against an AI system. In this section we'll describe a few and show how self-supervised training is less susceptible them.

The type of attack that can succeed with supervised trained ML systems without the adversary knowing anything about the ML system is called a black box attack. A black box attack tests the AI system with probes to determine how it will respond to various types of attacks. This type of attack is called an oracle attack.

The black box adversary collects datapoints by querying the "oracle" and builds a duplicate model based upon returns from the training set that is in actual use on the target. The objective of the legitimate training set is to observe data points and create a model that classifies each data point. In a security system the classifications may be simple: benign or malicious, the malicious classification presumably representing potential threats.

The adversary then duplicates the legitimate training set and slightly alters one or more of the datapoints so that an event that appears to be benign – and is classified as such – actually is not. These slight perturbations hark back to our examples of early AV tools. This is similar to slightly altering the signature of the virus without materially altering the virus. The AV misses it but the host is infected anyway. Knowing how the ML will respond, the adversary formulates her attacks accordingly.

In a supervised learning model this type of attack is feasible because the training set is static. However, that is

[5] Nicholas Carlini, "A Complete List of All (arXiv) Adversarial Example Papers" (https://nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html)

not the case with an self-supervised training model. In an self-supervised training model there is no static training set. The ML system learns from its environment. Therefore a black box query of the oracle will be unsuccessful because the system constantly is learning and there can be no misclassification.

Rules for the ML system to take its initial steps come in part from the algorithms that comprise its programming. We might think of these rules as, loosely, policies. Their purpose is to start the ML system on its learning path.

However, most of the Third Wave system's initial training actually comes from its observation of its environment. In simple terms, the ML system learns from observation how the host is supposed to behave and sets that as a baseline for its training.

V. LET'S ATTACK MIXMODE

While we have not subjected MixMode to actual attacks in the lab yet, we have run theoretical attacks to determine how we might expect the tool to react. To do that, we make some assumptions in formulating an attack and then look at how the tool is designed to respond. We assume a black box attack and a classification of benign, suspicious or malicious.

Our first example was a mass attack – sort of like a DDoS attack – against the oracle with most of the data likely to be acceptable and some small part not acceptable. Would MixMode pick out the unacceptable perturbations?

Since MixMode examines everything on the fly and already has the makings of a training model developed from its observation of acceptable behavior in the target system, it will pick out the perturbations intended to fool the classification system and classify them as malicious. This attack would not be expected to succeed on at least two levels.

First, the Third Wave ML will spot the perturbations immediately and report them. More important, perhaps, is that the attacker gets nothing back from the oracle. MixMode does not report out its decisions except to its cloud through secure channels. Thus, there are no perturbations for the attacker to use because he cannot tell how the MixMode classified.

Second, the perturbations are, at this point, guesses only and that type of attack is unreliable and produces no visible results to the attacker. So, the adversary has no idea what got past MixMode and what failed. The success factor for this attack is unacceptably (from the adversary's perspective) low.

Let's go to the other end of the spectrum. In this case we'll assume that our adversary has a way to see how MixMode responds. Perhaps he is a malicious insider or a bot that somehow was introduced into the system and is spying on the data stream.

In this case the adversary can query the oracle randomly and the insider can observe MixMode's responses. The adversary can, using knowledge of how the system is classifying, create a perturbation that will attempt to fool the classifier.

The problem with that goes back to our Third Wave approach of building off of the environment for a contextual-derived training set. The perturbation will make it into the system but MixMode will see it and note that it is abnormal. It will classify it as suspicious and watch for repeats. At this point, the system requires a bit of an assist from the human analyst to help it decide what is good and what is bad. It only suspects, for the present, that something might be amiss.

If it sees, repeats, and observes that the perturbation attempts actions that it, or its human analyst, knows to be malicious, it will change the classification to malicious and report it. In short, MixMode will not allow the malicious perturbation to become part of the acceptable environment.

If, on the other hand, the perturbation proves to offer no untoward activity it could become part of the acceptable environment. The end result? The virtual end of false positives and false negatives. In any event, this attack will not succeed.

VI. SUMMARY

In short, one of the adversary's goals is to attack the monitoring system – to, in effect, put out the defender's eyes. If we assume that the target network is not, itself, an AI-enabled system, if an attacker uses an AI-enabled attack it is likely that the attack will succeed without AI-enabled defenses in place.

Today, the state of the art in adversarial attacks – from real adversaries, not lab experiments – has not reached the level of a true AI-enabled attack. We do not yet have Third Wave malware that truly can exhibit emergent properties within a target network.

VII. SO, WHAT'S IN IT FOR ME?

The adversary's state of the practice is getting very close to Second Wave level adversarial attacks against the eyes of the defender. MixMode provides those eyes and is more than a generation ahead of the adversary. To our knowledge, MixMode is the only Third Wave AI network monitoring tool currently available.

But there are things about this type of artificial intelligence that definitely are to your advantage. First, you will be ready when the ability of humans to outperform intelligent attacks starts to end. Today, there are malware attacks and bot nets that can attack faster than current generation defenses can respond and much faster than humans can. Ransomware is an important example.

Second, managing an incident is not just about cleaning up the obvious mess made by the attacker, it's also about figuring out what happened and closing back doors. This is a network forensic process and Wave 3 AI is positioned to accomplish that quickly and reliably.

But it's not just the future – although that is a critical aspect of this type of monitoring tool. Today there are attacks – called fifth generation attacks[6] – that often can bypass current non-AI-driven tools. The ability of a Third Wave AI tool to understand that changes – however subtle – have been made in the enterprise and the data flowing on it is critical to detecting, blocking and analyzing these types of attacks.

As well, there is a misconception that AI learning takes days or even weeks or months to complete to a useful point. MixMode begins learning in the first few minutes, produces useful results in about an hour and can construct a complete enterprise baseline in a week. While we have not tested many aspects of MixMode in the lab yet, a long history of working in a production environment with the tool confirms this claim.

Finally, true Wave 3 AI is not available as far as we know in any other network monitoring tool so the ability for the adversary to blind the defenders increases as the sophistication of the "eyes" relative to attack sophistication decreases. For now and for the future – which is coming rapidly – Wave 3 self-supervised learning AI is a necessity.

[6] "... a large scale, multi-vector attack vector that is designed to infect multiple components of an information technology infrastructure, including networks, virtual machines, cloud instances and endpoint devices." See https://whatis.techtarget.com/definition/gen-V-attack)

www.mixmode.ai

+1 (858) 225-2352 | info@mixmode.ai | © 2020 MixMode, Inc.