# An Artificial Coevolutionary Framework for Adversarial AI

**Una-May O'Reilly** and **Erik Hemberg**
Massachusetts Institute of Technology

## Abstract

Cyber adversaries are engaged in a perpetual arms race. They are continuously maneuvering to outwit the opposing posture. Replicating and studying the dynamics of these engagements provides a route to proactive, adversarially-hardened cyber defenses. The constant struggle can be computationally formulated as a competitive coevolutionary system which generates many arms races that can be harvested for robust solutions. We present a paradigm, techniques and tools that recreate the coevolutionary process in the context of network cyber security scenarios. We describe its current use cases and how we harvest defensive solutions from it.

## Introduction

The greatest concern a prepared cyber defender might raise is: "What if my assumptions are wrong?" It is common knowledge that the only certainty is that an intelligent adversary will always keep trying to gain an advantage. Moreover, once forced to react, a defender is too late. So, how can a defender gain an edge in an environment that is stacked to the attacker's advantage, where the defender seems doomed to always be one step behind?

One approach is to deploy defensive configurations, that consider multiple possible *anticipated* adversarial behaviors and already take into account their expected impact, goals, strategies or tactics. Note that the precise metrics in this accounting can vary, For example, impact can be any combination of financial cost, disruption level or outcome risk. Or, a defender could prioritize a worst case, average case or a trade-off configuration.

One way that such configurations can be found is by using stochastic search methods that explore the simulated competitive behavior of adversaries and generate multi-ranked configurations from which a (human) defender can choose. In particular, the field of coevolutionary algorithms (Popovici et al. 2012) provides search heuristics that specifically direct competitive engagements between members of adversarial populations that undergo selection on the basis of performance with opposing objectives and variation to adapt. This logic results in population-wide adversarial dynamics. It can culminate in the possible adversarial behaviors that a defense would like to anticipate. For an example, see Figure 1

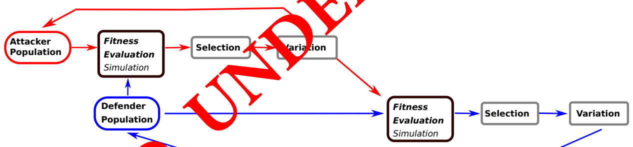

Figure 1: Alternating coevolutionary algorithm.

A competitive coevolutionary algorithm can be a component of a larger system, see Figure 2, in which a second component sets up the environment for adversarial engagement and measures the outcome for each adversary. These measures can be used by the coevolutionary algorithm to judge an adversary's fitness.

Herein we summarize a framework that we have used to generate robust defensive configurations (Prado Sanchez 2018; Pertierra 2018). It is composed of different coevolutionary algorithms that provide behavioral diversity. The algorithms, for further differentiation, use different "solution concepts", i.e. measures of adversarial success. Because engagements are frequently computationally expensive and have to be pairwise sampled from two populations each generation, the framework has a number of enhancements that enable more efficient use of a fixed budget of computation or time.

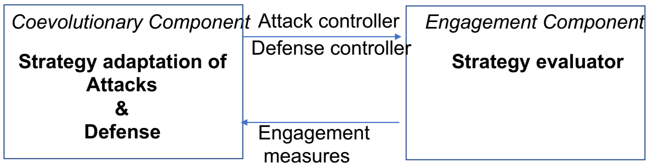

Figure 2: Component overview of our coevolutionary framework

The framework supports a number of use-cases using simulation and emulation of varying model granularity. These include: *A)* Defending a peer-2-peer network against Distributed Denial of Service (DDOS) attacks (Garcia et al. 2017) *B)* Defenses against spreading device compromise in a segmented enterprise network (Hemberg et al. 2018), and *C)* Deceptive defense against the internal reconnaissance of an adversary within a software defined network (Pertierra 2018) The framework is linked to a decision support module named `ESTABLO` (Sanchez et al. 2018; Prado Sanchez 2018). The engagements of every run of any of the coevolutionary algorithms are cached and, later, `ESTABLO` selects a subset of them for its *compendium* to combine adversaries resulting from different algorithms. It then competes the adversaries of each side against those of the other side and ranks each side's members according to multiple criteria. It also provides visualizations of adversarial behaviors and comparisons of their behavior. These "products" inform the decision process of a defensive manager.

The framework's novel contributions are:

- A coevolutionary algorithms system of diverse algorithms for preemptively investigating adversarial arms races and dynamics that could occur.
- Use cases modeling a variety of adversarial threats and defensive scenarios.
- A decision support module that supports selection of a superior anticipatory defensive configuration.

Section Background briefly summarizes related work. The coevolutionary method is described in Section Framework Methods. In Section Use-cases of coevolutionary algorithms in cyber security we provide examples of use-cases. Finally, a summary and future work are in Section Conclusions.

# Background

The strategy of testing the security of a system by trying to successfully attack it is akin to that which underlies fuzzing (Miller, Fredriksen, and So 1990). Just as software is tested by adaptive search for bugs, defenses can be tested by adaptive attacks. In contrast, while with software the bugs are fixed by humans and the base retested, in cybersecurity our intent is to additionally adapt defenses to counter attacks. Fuzzing is driven by genetic algorithms whereas we will drive cyber arms races in which both adversaries adapt using coupled GAs called competitive coevolutionary algorithms.

We describe some related work in modeling and simulation and coevolutionary algorithms.

## Modeling and Simulation

Modeling and simulation comprise a powerful approach, "*mod-sim*", for investigating general security scenarios (Tambe 2012) and computer security in particular (Thompson, Morris-King, and Cam 2016; Lange et al. 2017; Winterrose and Carter 2014). Mod-sim is often necessary because search and outcome spaces are too complex to derive analytical solutions while testbeds can incur long experimental cycle times and often do not abstract away irrelevant detail. Mod-sim systems range in complexity, level of abstraction and resolution.

## Coevolutionary Search Algorithms

Coevolutionary algorithms, related to evolutionary algorithms (Bäck 1996), explore domains in which the quality of a candidate *solution* is determined by its ability to successfully pass some set of *tests*. Reciprocally, a *test*'s quality is determined by its ability to force errors from some set of *solution*s. In competitive coevolution, similar to game theory, the search can lead to an arms race between *test* and *solution*, both evolving while pursuing opposite objectives (Popovici et al. 2012).

Coevolutionary algorithms can encounter problematic dynamics where *tests* are unable improve *solutions*, or drive toward a solution that is the *a priori* intended goal. There are accepted *remedies* to specific coevolutionary pathologies (Bongard and Lipson 2005; Ficici 2004; Popovici et al. 2012). They generally include maintaining population diversity so that a a search gradient is always present and using more explicit memory, e.g. a *Hall of Fame* or an archive, to prevent regress (Miconi 2009). The pathologies of coevolutionary algorithms are similar to those encountered by GANs (Goodfellow et al. 2014; Arora et al. 2017)

A related example to our framework in the domain of cyber security is CANDLES – the Coevolutionary, Agent-based, Network Defense Lightweight Event System (Rush, Tauritz, and Kent 2015). It is designed to coevolve attacker and defender strategies in the context of a single, custom, abstract computer network defense simulation.

# Framework Methods

## Coevolutionary Algorithms

A basic coevolutionary algorithm evolves two populations with e.g. tournament selection and for variation uses crossover and mutation. One population comprises attacks and the other defenses. In each generation, competitions are formed by pairing attack and defense. The populations are evolved in alternating steps: first the attacker population is selected, varied, updated and evaluated against the defenders, and then the defender population is selected, varied, updated and evaluated against the defenders. Each attacker–defender pair is dispatched to the engagement component to compete and the result is used as a component of fitness for each of them. Fitness is calculated over all an adversary's engagements.

The framework support diverse behavior through algorithms that vary in synchronization of the two populations and solution concepts. (Prado Sanchez 2018; Pertierra 2018). Working within a fixed time or fitness evaluation budget, the framework 1. caches engagements to avoid repeating them 2. uses Gaussian process estimation to identify and evaluate the most uncertain engagement (Pertierra 2018) 3. uses a recommender technique to approximate some adversary's fitnesses (Pertierra 2018) 4. uses a *spatial grid* to reduce complete pairwise engagements to a Moore neighborhood quantity(Mitchell 2006; Williams and Mitchell 2005).

### Decision support

Competitive coevolution has the following challenges (Sanchez et al. 2018; Prado Sanchez 2018): 1. Solutions and tests are not on comparable on a "level playing field" because fitness is based solely on the context of engagements. 2. Blind spots, unvisited by the algorithms may exist. 3. From multiple runs, with one or more algorithms, it is unclear how to automatically select a "best" solution.

Our decision support module, `ESTABLO`, see Figure 3, addresses these challenges. `ESTABLO`: 1. runs competitive coevolutionary search algorithms with different solution concepts. 2. combines the best solutions and tests at the end of each run into a compendium. 3. competes each solution against different test sets, including the compendium and a set of unseen tests, to measure its performance according to different solution concepts. 4. selects the "best" solutions from the compendium using a ranking and filtering process. 5. visualizes the best solutions to support a transparent and auditable decision.

## Use-cases of coevolutionary algorithms in cyber security

In computer security, guidance is sparse on how to prioritize or configure the many defensive postures, if it is available at all. In this section we demonstrate use-cases of how a coevolutionary algorithm framework can identify defensive configurations that are effective against a range of adversaries and scenarios in the attack kill-chain.

### DOS attacks on peer-to-peer networks

A peer-to-peer network is a robust and resilient means of securing mission reliability in the face of extreme DDOS attacks. The project named RIVALS (Garcia et al. 2017) assist in developing network defense strategies through modeling adversarial network attack and defense dynamics to help identify robust network design and deployment configurations that support mission completion despite an ongoing attack. Rather than manually tune and invent defense postures for a network every time an attacker adapts and acts, RIVALS assists during network design and hardening with the goal of anticipating attack evolution and identifying a robust defense that can circumvent the arms race and the reactive counter-measures. It uses coevolutionary algorithms to generate evolving network attacks and to evolve network defenses that effectively counter them.

RIVALS models DOS attack strategies by the attacker selecting one or more network servers to disable for some duration. Defenders can choose one of three different network routing protocols: shortest path, flooding and a peer-to-peer ring overlay to try to maintain their performance. Attack completion and resource cost minimization serve as attacker objectives. Mission completion and resource cost minimization are the reciprocal defender objectives. RIVALS' has a suite of coevolutionary algorithms that use archiving to maintain progressive exploration and that support different solution concepts as fitness metrics. Our experiments show that existing algorithms either sacrifice execution speed or forgo the assurance of consistent results.

### Availability attacks on segmented networks

Attackers do not always simply disrupt networks, instead they often introduce malware into networks. Once an attacker has compromised a device on a network, they can move to connected devices, akin to contagion. Here we consider *network segmentation*, a widely recommended defensive strategy, deployed against the threat of serial network security attacks that delay the mission of the network's operator (Hemberg et al. 2018).

We assume a network supports an enterprise in carrying out its business or *mission*, and that an adversary employs *availability attacks* against the network to disrupt this mission. Specifically, the attacker starts by using an exploit to compromise a vulnerable device on the network. This inflicts a mission delay when a mission critical device is infected. Then, the attacker moves laterally to compromise additional devices and maximally delay the mission.

Here, we examine a defensive measure called *network segmentation*, which divides the network topologically into *enclaves* that serve as isolation units to deter inter-enclave contagion. Network segmentation design is a tradeoff space: a more segmented network provides less mission efficiency because of increased overhead in inter-enclave communication. However, smaller enclaves contain compromise by limiting the spread rate, and their cleansing incurs fewer mission delays. Network operators can also use monitoring capabilities and network cleansing policies to detect and dislodge attackers.

We employ a simulation model to investigate the effectiveness over time of different defensive strategies against different attack strategies. The defender decides placement of mission devices and tap sensitivities in the enclaves. The attacker decides the strength, duration and number of attacks in a an attack plan. For a set of four network topologies, we generate strong availability attack patterns that were not identified *a priori*. Then, by combining the simulation with a coevolutionary algorithm to explore the adversaries' action spaces, we identify effective configurations that minimize mission delay when facing the attacks. The application of coevolutionary computation to enterprise network security represents a step toward course-of-action deter-

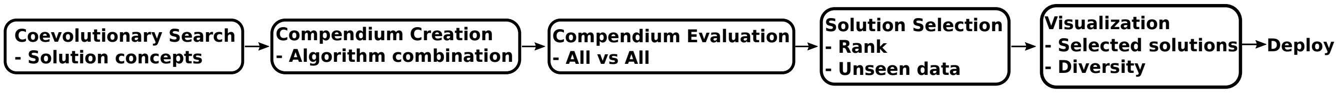

Figure 3: Overview of the ESTABLO framework for decision support through selection and visualization by using a compendium of solutions from coevolutionary algorithms.

mination that is robust to responses by intelligent adversaries.

### Internal reconnaissance in Software Defined Networks

Once an adversary has compromise a network endpoint, they perform reconnaissance of the network (Sood and Enbody 2013). After they have a view of the network and an understanding of where vulnerable nodes are located, they are able to execute a plan of attack. One way to protect against this is by obfuscating the network and delaying the attacker. A software defined network (SDN) can facilitate this, an SDN is a programmable network mostly used in cloud data centers (Kirkpatrick 2013). Thus, the SDN controller knows which machines are actually on the network and can control the network view for each machine, as well as place decoys(honeypots) on the network.

One such multi-component deceptive defense system (Achleitner, Laporta, and McDaniel 2016), foils scanning by generating “camouflaged” versions of the actual network and providing them to hosts when they renew their DHCP leases. We implemented a coevolutionary algorithm in order to explore the relationship between attacker and defender on a deceptive network (Pertierra 2018). This approach is used to discover optimal defender configurations to combat against malicious adversaries.

Using the deception system (Achleitner, Laporta, and McDaniel 2016), a modified POX SDN controller, we simulate a deceptive network with mininet (Team 2018). We run NMAP scans mimicing a node that is compromised and is performing reconnaissance on the network (Lyon 2018). The attacker behavior is: which IP addresses to scan, how many IP addresses to scan, which subnets to scan, the percent of the subnets to scan, the scanning speed, and the type of scan. The defender decisions are: the number of subnets to setup, the number of honeypots, the distribution of the real hosts throughout the subnets, and in our scenario the number of real hosts that exist on the network. The fitness scores are comprised of four components: how fast the defender detects that there is a scan taking place, the total time it takes to run the scan, the number of times that the defender detects the scanner, and the number of real hosts that the scanner discovers. Through experimentation and analysis, we discover certain configurations that the defender can use to significantly increase it's ability to detect entities that are scanning the network. Similarly, there are specific configurations that the attacking nodes can use to have a better chance of being undetected.

## Conclusion

We have described a paradigm that recreates the adversarial, competitive coevolutionary process in the domain of network cyber security scenarios in an abstract way. We presented its current use cases and how we harvest defensive solutions from it. Future work includes examining to more cyber security applications, more realistic engagements and more efficient algorithms.

## Acknowledgments

This material is based upon work supported by DARPA. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements. Either expressed or implied of Applied Communication Services, or the US Government.

## References

[Achleitner, Laporta, and McDaniel 2016] Achleitner, S.; Laporta, T.; and McDaniel, P. 2016. Cyber deception: Virtual networks to defend insider reconnaissance. *In Proceedings of the 2016 International Workshop on Managing Insider Security Threats* 57–68.

[Arora et al. 2017] Arora, S.; Ge, R.; Liang, Y.; Ma, T.; and Zhang, Y. 2017. Generalization and equilibrium in generative adversarial nets (gans). *arXiv preprint arXiv:1703.00573.*

[Bäck 1996] Bäck, T. 1996. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms.* Oxford University Press.

[Bongard and Lipson 2005] Bongard, J. C., and Lipson, H. 2005. Nonlinear system identification using coevolution of models and tests. *IEEE Transactions on Evolutionary Computation* 9(4):361–384.

[Ficici 2004] Ficici, S. G. 2004. *Solution concepts in coevolutionary algorithms.* Ph.D. Dissertation, Citeseer.

[Garcia et al. 2017] Garcia, D.; Lugo, A. E.; Hemberg, E.; and O'Reilly, U.-M. 2017. Investigating coevolutionary archive based genetic algorithms on cyber defense networks. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '17, 1455–1462. New York, NY, USA: ACM.

[Goodfellow et al. 2014] Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2672–2680.

[Hemberg et al. 2018] Hemberg, E.; Zipkin, J. R.; Skowyra, R. W.; Wagner, N.; and O'Reilly, U.-M. 2018. Adversarial co-evolution of attack and defense in a segmented computer network environment. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 1648–1655. ACM.

[Kirkpatrick 2013] Kirkpatrick, K. 2013. Software-defined networking. *Communications of the ACM* 56(9).

[Lange et al. 2017] Lange, M.; Kott, A.; Ben-Asher, N.; Mees, W.; Baykal, N.; Vidu, C.-M.; Merialdo, M.; Malowidzki, M.; and Madahar, B. 2017. Recommendations for model-driven paradigms for integrated approaches to cyber defense. *arXiv preprint arXiv:1703.03306*.

[Lyon 2018] Lyon, G. 2018. Nmap network scanner. `https://nmap.org/`. [Online; accessed 6-July-2018].

[Miconi 2009] Miconi, T. 2009. Why coevolution doesn't "work": superiority and progress in coevolution. In *European Conference on Genetic Programming*, 49–60. Springer Berlin Heidelberg.

[Miller, Fredriksen, and So 1990] Miller, B. P.; Fredriksen, L.; and So, B. 1990. An empirical study of the reliability of unix utilities. *Communications of the ACM* 33(12):32–44.

[Mitchell 2006] Mitchell, M. 2006. Coevolutionary learning with spatially distributed populations. *Computational intelligence: principles and practice*.

[Pertierra 2018] Pertierra, M. 2018. Investigating coevolutionary algorithms for expensive fitness evaluations in cybersecurity. Master's thesis, Massachusetts Institute of Technology.

[Popovici et al. 2012] Popovici, E.; Bucci, A.; Wiegand, R. P.; and De Jong, E. D. 2012. Coevolutionary principles. In *Handbook of natural computing*. Springer. 987–1033.

[Prado Sanchez 2018] Prado Sanchez, D. 2018. Visualizing adversaries - transparent pooling approaches for decision support in cybersecurity. Master's thesis, Massachusetts Institute of Technology.

[Rush, Tauritz, and Kent 2015] Rush, G.; Tauritz, D. R.; and Kent, A. D. 2015. Coevolutionary agent-based network defense lightweight event system (candles). In *Proceedings of the Companion Publication of the 2015 on Genetic and Evolutionary Computation Conference*, 859–866. ACM.

[Sanchez et al. 2018] Sanchez, D. P.; Pertierra, M. A.; Hemberg, E.; and O'Reilly, U.-M. 2018. Competitive coevolutionary algorithm decision support. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 300–301. ACM.

[Sood and Enbody 2013] Sood, A., and Enbody, R. 2013. Targeted cyberattacks: a superset of advanced persistent threats. *IEEE security & privacy* 11(1):54–61.

[Tambe 2012] Tambe, M., ed. 2012. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press.

[Team 2018] Team, M. 2018. Mininet - realistic virtual sdn network emulator. `http://mininet.org/`. [Online; accessed 6-July-2018].

[Thompson, Morris-King, and Cam 2016] Thompson, B.; Morris-King, J.; and Cam, H. 2016. Controlling risk of data exfiltration in cyber networks due to stealthy propagating malware. In *Military Communications Conference, MILCOM 2016-2016 IEEE*, 479–484. IEEE.

[Williams and Mitchell 2005] Williams, N., and Mitchell, M. 2005. Investigating the success of spatial coevolution. In *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, 523–530. ACM.

[Winterrose and Carter 2014] Winterrose, M. L., and Carter, K. M. 2014. Strategic evolution of adversaries against temporal platform diversity active cyber defenses. In *Proceedings of the 2014 Symposium on Agent Directed Simulation*, 9. Society for Computer Simulation International.