
FIXED POINTS IN CYBER SPACE: RETHINKING OPTIMAL EVASION ATTACKS IN THE AGE OF AI-NIDS

Christian Schroeder de Witt*
Department of Engineering Science
University of Oxford
cs@robots.ox.ac.uk

Yongchao Huang
Department of Computer Science
University of Oxford
yongchao.huang@cs.ox.ac.uk

Phil H.S. Torr
Department of Engineering Science
University of Oxford
philip.torr@eng.ox.ac.uk

Martin Strohmeier
armasuisse Science and Technology
Thun, Switzerland
martin.strohmeier@ar.admin.ch

ABSTRACT

Cyber attacks are increasing in volume, frequency, and complexity. In response, the security community is looking toward fully automating cyber defense systems using machine learning. However, so far the resultant effects on the coevolutionary dynamics of attackers and defenders have not been examined. In this whitepaper, we hypothesise that increased automation on both sides will accelerate the coevolutionary cycle, thus begging the question of whether there are any resultant fixed points, and how they are characterised. Working within the threat model of Locked Shields, Europe’s largest cyberdefense exercise, we study blackbox adversarial attacks on network classifiers. Given already existing attack capabilities, we question the utility of optimal evasion attack frameworks based on minimal evasion distances. Instead, we suggest a novel reinforcement learning setting that can be used to efficiently generate arbitrary adversarial perturbations. We then argue that attacker-defender fixed points are themselves general-sum games with complex phase transitions, and introduce a temporally extended multi-agent reinforcement learning framework in which the resultant dynamics can be studied. We hypothesise that one plausible fixed point of AI-NIDS may be a scenario where the defense strategy relies heavily on whitelisted feature flow subspaces. Finally, we demonstrate that a continual learning approach is required to study attacker-defender dynamics in temporally extended general-sum games.

1 INTRODUCTION

Driven by ever-increasing infrastructure growth, cyber attacks are becoming increasingly frequent, sophisticated, and concealed. The use of semi-automated network intrusion detection systems using machine learning technology (ML-NIDS) reduces the workload on human network operators. However, modern cyber attacks are in turn increasingly drawing on machine learning techniques, thus perpetuating the continual arms race between cyber attackers and defenders.

To mitigate the natural asymmetry between attackers and defenders, recent work (Meier et al., 2021) has, possibly for the first time, sought to described a fully-automated network defense system. We will in the following refer to such systems as *AI-NIDS*, in order to distinguish human-in-the-loop systems relying on machine learning technology (*ML-NIDS*) from systems that make decisions in a fully autonomous fashion.

While the concept of AI-NIDS may seem intriguing to practitioners, to the best of our knowledge, so far nobody has studied their effect on the coevolutionary attacker-defender dynamics. This is surprising given the potentially game-changing implications of interacting, fully automated cyber attack and defense systems in the near future: While the speed of today’s cyber security arms race is mostly limited by human ingenuity, automated systems of the future will interact and coevolve on significantly smaller timescales, learning new exploits and defenses against over the course of not months, but hours, or even milliseconds. Analogous escalations due to increased automation are starting to be observed in financial markets (Alexandrov et al., 2021).

*Corresponding author. Please contact at cs@robots.ox.ac.uk

In this paper, we argue that the advent of AI-NIDS may trigger a coevolutionary explosion with wide-reaching consequences for cyber security practice. In this new era of nearly-instantaneous attacker-defender coevolution, we anticipate that the study of mutual best responses between fixed populations of attack and defense techniques will have to be replaced by a better understanding of the emerging dynamical fixed points that the attacker-defender systems might evolve toward.

Without loss of generality, we in this work focus on system dynamics arising from network-classifier based defenses against botnet infiltration. Botnets pose an increasingly severe security threat to the global security environment, enabling crimes such as information theft and distributed denial-of-service attacks Abu Rajab et al. (2006). Network flow classification (NFC) is a central security component in botnet detection systems (Abraham et al., 2018). In traditional network intrusion detection systems (IDS), classifier rules are little more than codified expert knowledge based on a set of known malicious and benign behaviour. With the rapid growth of networking systems, ever-evolving attack patterns and the rising complexity of communication protocols, state-of-the-art NFC techniques increasingly rely on machine learning (ML) to train parameterised classifiers directly from large amounts of labeled examples (Lakhina et al., 2004). In fact, the automated cyber defense system proposed by Meier et al. (2021) makes heavy use of machine learning-based NFCs.

While being promising in many practical applications (Wagner & Soto, 2002), it has been shown that ML-based classifiers are inherently vulnerable to adversarial attacks (Dalvi et al., 2004). One such class of attacks includes the poisoning of training data pools (Barreno et al., 2006). Another such class of attacks are optimal evasion attacks (Nelson et al., 2010), in which an attacker changes its communication patterns such as to avoid NFC detection. In this paper, we restrict ourselves to the study of so-called *blackbox* optimal evasion attacks, referring to settings in which attackers and defenders do not have access to each other’s internal models but can solely infer the other’s state through real-world interactions. Adversarial blackbox attacks on network flow classifiers have recently been demonstrated using reinforcement learning with a sparse feedback signal in order to generate adversarial perturbations allowing malicious communication to be masked (Apruzzese et al., 2020).

While being ubiquitously studied due their satisfying mathematical properties, optimal evasion frameworks generally neglect a crucial characteristic of real-world NFC evasion attacks. In fact, radically altering malicious traffic distributions in stochastic flow feature space does not necessarily impose any tangible costs on the attacker¹: An abundance of specialised networking software (Biondi & Community, 2021, Scapy), compression, encryption, and myriads of openly available repositories of wildly diverse C&C communication protocols² mean that attackers do not need to search for the smallest possible adversarial perturbations. Instead, as long as overall information throughput is guaranteed to remain above a certain operational threshold - that is usually small compared to benign network traffic -, we hypothesise that real-world attackers can traverse statistical flow feature (SFF) space almost arbitrarily at little cost. With the advent of increasingly powerful AI-based program synthesis methods (Church, 1963; Waldinger & Lee, 1969; Johnson et al., 2017; Parisotto et al., 2016; Devlin et al., 2017), we expect that future attackers will be able to generate efficient C&C communication protocols with associated statistical flow feature properties *a la carte*, and at negligible cost.

The framework introduced by Apruzzese et al. (2020) does not readily extend to settings in which adversarial perturbations of large or arbitrary sizes need to be generated as each feature modification requires a full environment step. Very long episodes are not only inefficient to generate, but can also cause issues with temporal credit assignment during training. We thus introduce *Fast Adversarial Sample Training* (FAST), a reinforcement learning method for blackbox adversarial attacks that can generate continuous perturbations of arbitrary size in just a single step. We show that FAST can be efficiently used to generate discrete features. For completeness, we show FAST to be efficient and easily tunable even in minimal evasion distance settings.

Defense against NFC blackbox attacks is traditionally achieved by *hardening*, i.e. by supervised training on known (or artificially generated) malicious perturbations. In fact, within the Apruzzese et al. (2020) framework, we demonstrate that, under favourable conditions, NFCs can be hardened sufficiently in just one retraining step. This means that subsequent attacks using the same method will largely fail to deceive the hardened classifier within over the same flow sample distribution. We thus present empirical evidence that, under certain conditions, blackbox minimal distance evasion attack dynamics, can indeed converge to a no-attack fixed point. However, once the minimal evasion distance framework is abandoned, such fixed points are clearly unlikely to be stable.

Instead, we argue that understanding the dynamics of *arbitrary distance evasion attacks* (ADEA) requires modeling the attacker-defender system as a *temporally extended general-sum game*. We note that the game is not *zero-sum*, as may

¹This observation was made already in Merkli (2020), but remained largely inconsequential.

²Try <https://www.google.com/search?q=github+command+and+control+protocols>.

commonly be assumed, because, while attackers try to infect nodes and defenders try to keep from being infected, both attackers and defenders are simultaneously interested in keeping network services operational: If an attacker compromises network operations to a noticeable extent, then the risk of security escalations or even network shutdowns in response increases - this in turn compromises botnet operations. In return, defenders might worry less about possible infections as long as network operations remain unaffected. The goals of attackers and defenders are thus partially aligned. We proceed to formulate such games as a deep multi-agent reinforcement learning problem, CyberMARL, in which both attackers and defenders share the same network channel. CyberMARL easily accommodates for complications found in real-world NIDS operations, such as delayed feedbacks (Apruzzese et al., 2021). In addition, in CyberMARL, we explicitly assume NFCs to operate concurrently to other ML-NIDS techniques, such as anomaly detectors and blacklisting. We illustrate the utility of CyberMARL in modeling a *mode transition*, i.e. a period of time during which the attacker suddenly changes attack distributions, thus pushing the defender NFC temporarily out-of-distribution.

How could cyber defense systems of the distant future defend themselves effectively from ADEAs? We hypothesize that one plausible fixed point of attacker-defender coevolution may be a scenario we dub *Whitelisting Hell*. In *Whitelisting Hell*, AI-NIDS automatically drop network traffic whose SFFs lie outside of very narrow, temporally changing, whitelisted subspaces (or "corridors"). In order to be able to create and operate a botnet under such conditions, the attacker needs to predict the dynamics of the whitelisted subspaces in order to generate C&C flows within these.

Whether in *Whitelisting Hell*, or under the more general ADEA conditions explored by CyberMARL, possible attack strategies may seek to exploit a weakness of supervised classifiers: Neural networks, as well as other classifiers, suffer from catastrophic forgetting. As an antidote, training samples could be exhaustively stored and the classifier be retrained continuously, however, this may not be possible given traffic volumes, as well as real-time constraints and computation budgets even of large-scale AI-NIDS systems. We suggest and empirically illustrate that NFC defenses in future AI-NIDS may benefit from *continual learning* (Parisi et al., 2019) techniques.

This paper proceeds by first providing the necessary background on botnet detection, network classifiers, Locked Shields, blackbox adversarial attacks, and deep multi-agent reinforcement learning (Section 2). We subsequently introduce and evaluate FAST (Section 3), followed by CyberMARL (Section 4), and conclude with a description of *Whitelisting Hell*, and an empirical illustration of the demonstration of continual learning techniques in temporally extended attacker-defender settings (Section 5).

2 BACKGROUND

2.1 BOTNETS, NETWORK FLOW CLASSIFIERS AND LOCKED SHIELDS

Botnets pose an increasingly severe security threat to the global security environment, enabling crimes such as information theft and distributed denial-of-service attacks (Abu Rajab et al., 2006). Network flow classification (NFC) is a central security component in botnet detection systems (Abraham et al., 2018). In NFC, a blue team classifier decides whether a given intercepted network package is benign or part of team red's botnet communication stream, for example the communication between an infected host and a command and control (C&C) server (see Figure 1). Packages that are identified as malicious may simply be dropped by team blue, thus blocking team red's C&C channels.

In traditional network intrusion detection systems (IDS), classifier rules codified expert knowledge based on a set of known malicious and benign behaviour. With the rapid growth of networking systems, ever-evolving attack patterns and the rising complexity of communication protocols, state-of-the-art NFC techniques increasingly rely on machine learning (ML) to train parameterised classifiers directly from large amounts of labeled examples (Lakhina et al., 2004).

While being promising in many practical applications (Wagner & Soto, 2002), it has been shown that ML-based classifiers are inherently vulnerable to adversarial attacks (Dalvi et al., 2004). One such class of attacks includes the poisoning of training data pools (Barreno et al., 2006). Another such class of attacks are optimal evasion attacks (Nelson et al., 2010), in which an attacker changes its communication patterns such as to avoid NFC detection.

Locked Shields (LS) is one of the world's largest cyber defense exercises, being held annually in Tallinn, Estonia (CCDCOE, 2020). In LS, Team Red (R) attempts to infiltrate internal network nodes belonging to Team Blue (B) in order to establish a botnet. While team red is supplied with many details about the network and allowed to infiltrate it ahead of time, team blue only has band-width limited access to a few nodes and tight computational budget constraints. Team Blue's goal consists of identifying infiltrated nodes and interrupt the operation of Team Red's botnet.

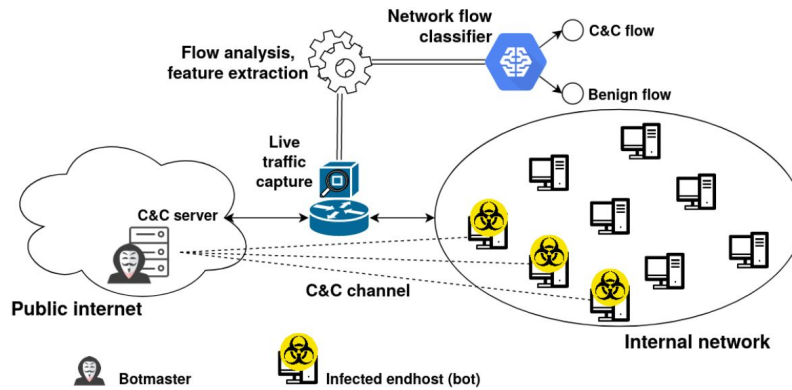


Figure 1: Network flow classification for detection of C&C channels. Taken from Merkli (2020)

Statistical Flow Features Statistical flow features (SFF), such as those extracted by CICFlowMeter (Lashkari, 2020), are based entirely on package meta-information, not content, meaning they can be evaluated even for encrypted or compressed flows. Despite - or indeed because of - this level of abstraction, SFFs have been found to be useful features for a variety of cyber security tasks.

Adversarial attacks and defenses Deep Neural Networks (DNNs) are known to be vulnerable to adversarial perturbations (Ren et al., 2020). An *adversarial perturbation* δ is a small term that, when added to a benign sample x with classifier output label y , results in an adversarial sample that is both realistic, but results either in a different classifier output label (untargeted) or a maliciously chosen one (targeted). Many methods for generating such adversarial samples try to induce visual realism by minimizing a pixel-space L_p norm between the adversarial sample and the benign sample (Carlini & Wagner, 2017; Liu et al., 2017). However, such a realism criterion is problematic in flow feature space (Merikli, 2020).

Optimal evasion attacks An *optimal evasion* attack is an adversarial attack using a perturbation that is optimal with respect to some optimality criteria. Although in principle, this optimality criterion could be chosen freely, recent work has almost exclusively focused on evasion attacks where the size of the adversarial perturbation generated is to be minimised: in other words, attackers are assumed to incur a cost proportional to the size of the adversarial perturbation. Formally, such *minimal distance evasion attacks (MDEA)*, see Figure 2 (Merikli, 2020) posit that, given some binary discriminator D_{rf} , the goal is to identify perturbations δx_i such that $D_{rf}(x_i + \delta x_i) = \neg D_{rf}(x_i)$ while minimizing $\|\delta x_i\|_p^2$, $p \geq 1$. Merikli (2020) employ a random forest discriminator and find δx_i by solving a MILP problem first proposed by (Kantchelian et al., 2016), using an a priori fixed set of N labeled samples $x_i \in \mathcal{X}$.

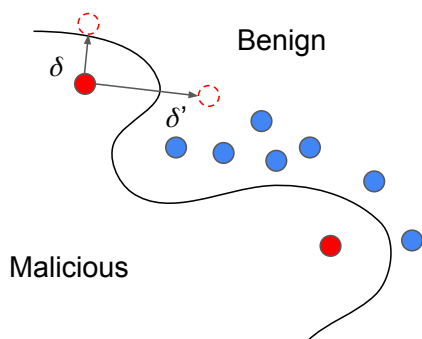


Figure 2: An illustration of minimal distance evasion attacks.

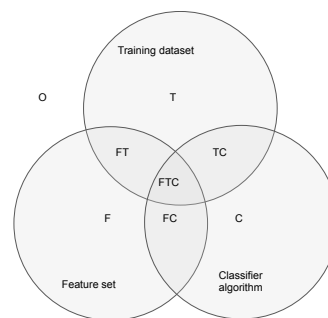


Figure 3: Levels of access. Taken from (Merikli, 2020).

Blackbox attacks Adversarial attacks may occur with various levels of access (see Figure 3). Attackers might, for example, have access to the defender’s NFC model weights, and/or the classifier features, or the defender’s training set. In this paper, we focus on so-called *blackbox* settings in which the attacker cannot be assumed to have access to any of these

2.2 REINFORCEMENT LEARNING AND GENERAL-SUM GAMES

Partially-Observable Stochastic Games (POSGs Hansen et al., 2004) describe multi-agent tasks where a number of agents with individual goals choose sequential actions under partial observability and environment stochasticity. POSGs can be formally defined by a tuple $\langle \mathcal{N}, \mathcal{S}, \mathcal{U}, P, r_a, \mathcal{Z}, O, \rho, \gamma \rangle$. Here $s \in \mathcal{S}$ describes the state of the environment, discrete or continuous, and $\mathcal{N} := \{1, \dots, N\}$ denotes the set of N agents. $s_0 \sim \rho$, the initial state, is drawn from distribution ρ . At each time step t , all agents $a \in \mathcal{N}$ simultaneously choose actions $u_t^a \in \mathcal{U}$ which may be discrete or continuous. This yields the joint action $\mathbf{u}_t := \{u_t^a\}_{a=1}^N \in \mathcal{U}^N$. The next state $s_{t+1} \sim P(s_t, \mathbf{u}_t)$ is drawn from transition kernel P after executing the joint action \mathbf{u}_t in state s_t . Subsequently, agent a receive a scalar reward $r_t^a = r^a(s_t, \mathbf{u}_t^a)$.

Instead of being able to observe the full state s_t , in a POSG each agent $a \in \mathcal{N}$ can only draw an individual local observation $z_t^a \in \mathcal{Z}$, $\mathbf{z}_t := \{z_t^a\}_{a=1}^N$, from the observation kernel $O(s_t, a)$. The history of an agent’s observations and actions is denoted by $\tau_t^a \in \mathcal{T}_t := (\mathcal{Z} \times \mathcal{U})^t \times \mathcal{Z}$. The set of all agents’ histories is given by $\tau_t := \{\tau_t^a\}_{a=1}^N$. Each agent a chooses its actions with a decentralised policy $u_t^a \sim \pi^a(\cdot | \tau_t^a)$ that is based only on its individual history.

Each agent attempts to learn an individual *policy* $\pi^a(u^a | \tau_t^a)$ that maximises the agent’s expected discounted return, $J(\pi^a) \doteq \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t^a]$, where $\gamma \in [0, 1)$ is a discount factor. $\pi^a(u^a | \tau_t^a)$ induces an individual action-value function $Q_a^\pi := \mathbb{E}[\sum_{i=0}^{\infty} \gamma^i r_{t+i}^a]$ that estimates the expected discounted return of agent a ’s action u_t^a in state s_t with agent history τ_t .

Centralised learning with decentralised execution. Reinforcement learning policies can often be learnt in simulation or in a laboratory. In this case, on top of their local observation histories, agents may have access to the full environment state and share each other’s policies and experiences during training. The framework of *centralised training with decentralised execution (CTDE)* (Oliehoek & Amato, 2016; Kraemer & Banerjee, 2016) formalises the use of centralised information to facilitate the training of decentralisable policies.

Deep Q-learning *Deep Q-Network* (DQN) (Mnih et al., 2015) uses a deep neural network to estimate the action-value function, $Q(s, \tau, \mathbf{u}; \theta) \approx \max_{\pi} Q^\pi(s, \tau, \mathbf{u})$, where θ are the parameters of the network. For the sake of simplicity, we assume here feed-forward networks, which condition on the last observations \mathbf{z}_t^a , rather than the entire agent histories τ_t . The network parameters θ are trained by gradient descent on the mean squared regression loss:

$$\mathcal{L}_{[\theta]}^{\text{DQN}} := \mathbb{E}_{\mathcal{D}} \left[\left(y_t^{\text{DQN}} - Q(s_t, \mathbf{z}_t, \mathbf{u}_t; \theta) \right)^2 \right], \quad (1)$$

where $y_t^{\text{DQN}} := r_t + \gamma \max_{\mathbf{u}'} Q(s_{t+1}, \mathbf{z}_{t+1}, \mathbf{u}'; \theta^-)$ and the expectation is estimated with transitions $(s_t, \mathbf{z}_t, \mathbf{u}_t, r_t, s_{t+1}, \mathbf{z}_{t+1}) \sim \mathcal{D}$ sampled from an *experience replay buffer* \mathcal{D} (Lin, 1992). The use of replay buffer reduces correlations in the observation sequence. To further stabilise learning, θ^- denotes parameters of a *target network* that are only periodically copied from the most recent θ . While we do not employ DQN in this paper, it is an important reference algorithm in reinforcement learning (Sutton & Barto, 2018). Independent Q-learning (IQL) is a simple extension of single-agent Q-learning to multi-agent settings (Tan, 1993).

Deep Deterministic Policy Gradients (DDPG) Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al., 2015) is an actor-critic algorithm that poses an important alternative to continuous Q-learning (Gu et al., 2016). In DDPG, an actor has a deterministic policy μ parametrised by θ . The actor is learnt alongside a critic, Q that conditions on the agent’s observation (or the full environment state s , if centralised training is available). The critic is updated using a TD-error loss:

$$\mathcal{L}_{[\phi]}^{\text{DDPG}} := \mathbb{E}_{\mathcal{D}} \left[\left(y_t - Q^\mu(s_t, \mathbf{u}_t; \phi) \right)^2 \right], \quad (2)$$

where $y_t := r_t + \gamma Q^\mu(s_{t+1}, \mu(\tau_{t+1}; \theta'); \phi')$ and transitions are sampled from a replay buffer \mathcal{D} (Lin, 1992) and θ' and ϕ' are target-network parameters.

Multi-Agent Deep Deterministic Policy Gradient (MADDPG) *Multi-agent deep deterministic policy gradient* (MADDPG Lowe et al., 2017) is an actor-critic method that works in both cooperative and competitive MARL tasks with discrete or continuous action spaces. MADDPG was originally designed for the general case of *partially*

observable stochastic games (Kuhn, 1953), in which it learns a separate actor and centralised critic for each agent such that agents can learn arbitrary reward functions - including conflicting rewards in competitive settings. In this paper, we employ a variant of MADDPG with continuous action spaces. We assume each agent a has a deterministic policy μ^a , parameterised by θ^a , with $\boldsymbol{\mu}(\boldsymbol{\tau}; \boldsymbol{\theta}) := \{\mu^a(\tau^a; \theta^a)\}_{a=1}^N$. For POSGs, MADDPG learns individual centralised critics $Q_a^\mu(s, \mathbf{u}; \phi)$ for each agent a with shared weights ϕ that condition on the full state s and the joint actions \mathbf{u} of all agents. The policy gradient for θ^a is given by:

$$\nabla_{\theta^a} \mathcal{L}_{[\theta^a]}^\mu := -\mathbb{E}_{\mathcal{D}} \left[\nabla_{\theta^a} \mu^a(\tau_t^a; \theta^a) \nabla_{\mathbf{u}^a} Q_a^\mu(s_t, \hat{\mathbf{u}}_t^a; \phi) \Big|_{\mathbf{u}^a = \mu^a(\tau_t^a)} \right],$$

where $\hat{\mathbf{u}}_t^a := \{u_t^1, \dots, u_t^{a-1}, u_t^a, u_t^{a+1}, \dots, u_t^N\}$ and $s_t, \mathbf{u}_t, \tau_t$ are sampled from a replay buffer \mathcal{D} . The shared centralised critic Q_a^μ is trained by minimising the following loss:

$$\mathcal{L}_{[\phi]}^{\text{DPG}} := \mathbb{E}_{\mathcal{D}} \left[\left(y_t^a - Q_a^\mu(s_t, \mathbf{u}_t; \phi) \right)^2 \right], \quad (3)$$

where $y_t^a := r_t + \gamma Q_a^\mu(s_{t+1}, \boldsymbol{\mu}(\boldsymbol{\tau}_{t+1}; \boldsymbol{\theta}'); \phi')$ and transitions are sampled from a replay buffer \mathcal{D} (Lin, 1992) and $\boldsymbol{\theta}'$ and ϕ' are target-network parameters.

2.3 BLACKBOX ADVERSARIAL ATTACKS

Network Flow Classifiers (NFCs) are mappings $C : \mathcal{F} \rightarrow [0, 1]$ that take a sample $x \in \mathcal{F}$, where \mathcal{F} is the feature space, to a score $\rho(x)$. Scores ≥ 0.5 indicate a malicious sample, scores below indicate a benign sample. Given a sample x_{adv} for which $C(x_{adv}) < 0.5$, the task of an adversarial sample generator is then to generate a perturbation δ such that $C(x_{adv} + \delta) \geq 0.5$ ³. Throughout this paper, C corresponds to a fixed classifier trained according to section 3 using a random class-balanced set of 10^6 samples from 2018 Locked Shields data.

While neural network architectures are starting to catch up (Chernikova & Oprea, 2020; Apruzzese et al., 2020), state-of-the-art classifier models are generally based on random forests (Merkli, 2020). Kantchelian et al. (2016) introduce a specialised evasion attack generation scheme for random forests that requires the costly solution of a mixed integer linear program (MILP) for each adversarial sample. Not only greatly does this limit the rate of adversarial samples that can be generated, but, in addition, the inclusion of side constraints - e.g. to ensure that the generated samples are realistic - is greatly limited. As random forests are not end-to-end differentiable, approaches employing generative adversarial schemes are not immediately applicable (Goodfellow et al., 2014).

Apruzzese et al. (2020) introduce a novel reinforcement learning-based method for generating adversarial perturbations in blackbox settings. Their method employs Q-learning over episodes with step-wise feature modifications (see Figure 4). While empirically shown to be able to generate efficient attacks, Apruzzese et al. (2020)'s method suffers from potentially extremely long episode lengths if perturbation sizes are substantial, making it practically inapplicable outside of MDEA settings. Even within MDEA settings, the episodic nature of perturbation generation may impede the learning process as it may introduce temporal credit-assignment issues.

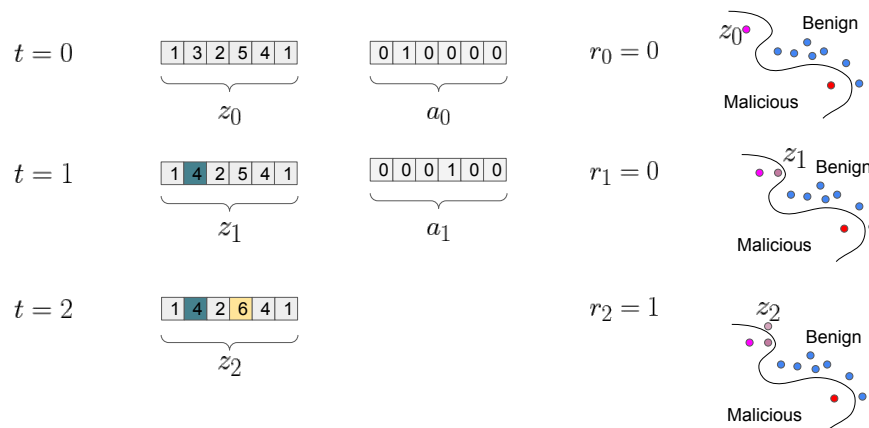


Figure 4: An illustration of (Apruzzese et al., 2020).

³Such a perturbation is likely not unique.

3 OPTIMAL EVASION WITH ARBITRARY SINGLE-STEP PERTURBATIONS

To overcome the limitations of adversarial sample generation in blackbox settings posed by Apruzzese et al. (2020)’s method, we now introduce a novel setting in which, discrete or continuous, perturbations of arbitrary size are generated in a single step.

Fast Adversarial Sample Training (FAST). (FAST) formulates the process of finding a suitable δ as a single-agent reinforcement learning problem as follows (see Figure 5): An attacker with policy π^A receives an i.i.d. random sample $z \sim \mathcal{D}_{adv}$ from the environment \mathcal{E} . The attacker then constructs a perturbation $\delta := \pi^A(z)$ and receives a reward that is +1 if the NFC has been fooled, i.e. $C(x + \delta) < 0.5$ and 0.0 otherwise. In contrast to Apruzzese et al. (2020), we employ deep deterministic policy gradients (DDPG) (see Section 2) and generate continuous perturbations, which are discretised post-hoc if feature spaces are discrete. Each episode terminates after a single step.

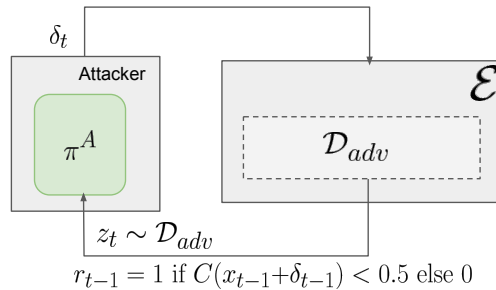


Figure 5: Schematic depiction of FAST’s reinforcement learning loop, where \mathcal{E} denotes the environment.

To illustrate FAST’s versatility even in MDEA settings, we evaluate FAST on a set of 10^6 class-balanced i.i.d. sampled sets of Locked Shields samples \mathcal{D}_{ben} , \mathcal{D}_{mal} from the year 2018. We train the attacker policy π using DDPG (see section 2.2), using three-layered fully connected neural network architecture with tanh-activation functions and 64 activations per layer for both actor and critic (see Figure 15). We fix our actor learning rate at 10^{-4} , and our critic learning rate at 10^{-3} and choose the discount factor $\gamma = 0.99$. To incentivise small perturbations, we add a regulariser the L2 norm $\|\delta\|_2$. Note that perturbation constraints are feature-wise and scaled to feature-wise normalised features - absolute perturbations can be derived by first multiplying δ by the standard deviation of the respective feature and then adding its mean (see Table 5).

We empirically investigate the *attack rate*, i.e. the ratio of generated perturbations that fool the classifier, after 10^6 training steps. We find that FAST is able to fool the classifier about 15% of the time even if we restrict feature-wise perturbations to 1/100th of the feature standard deviation. If we increase the allowed perturbations by another factor 10, then we reach almost perfect attack rates (see Figure 1). We leave detailed empirical comparisons of FAST with Apruzzese et al. (2020)’s method for future work as our methodological advances are clear, and further empirical investigation lies beyond the scope of our paper.

Year \ $\ \delta\ $	0.001	0.003	0.01	0.03	0.1
2018	0.84%	1.2%	14.9%	63.6%	97.1%

Table 1: Attack rate for FAST given different hard upper constraints on the perturbation L2-norm $\|\delta\|_2$. The LightGBM model has been trained on the top 20 most important features from class-balanced random samples from 2018 Locked Shields data. Note that the perturbation norm is expressed relative to feature-wise normalised samples.

4 FROM OPTIMAL EVASION ATTACKS TO TEMPORALLY-EXTENDED GENERAL-SUM GAMES

4.1 ITERATIVE HARDENING

Defense against NFC blackbox attacks is traditionally achieved by hardening, i.e. by supervised training on known (or artificially generated) malicious perturbations. We now demonstrate that, within the Apruzzese et al. (2020) framework, under favourable conditions, NFCs can be hardened sufficiently in just one retraining step. This means that subsequent attacks using the same method will largely fail to deceive the hardened classifier within over the same flow sample

distribution. We thus present empirical evidence that, under certain conditions, blackbox MDEA dynamics, can indeed converge to a no-attack fixed point (see Figure 6). However, within the MDEA framework, such fixed points are clearly unlikely to be stable as the attacker can choose to increase perturbation sizes at no cost.

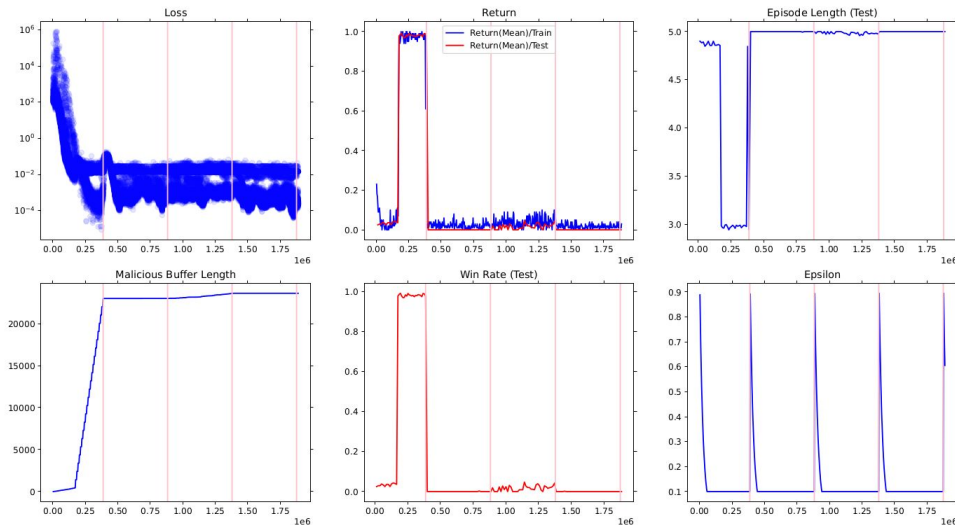


Figure 6: Iterative Hardening in the Q-learning framework introduced by Apruzzese et al. (2020). We find that already after a single cycle of retraining on malicious perturbations, the classifier is hardened against future 5-step exploits.

4.2 CYBERMARL

To investigate the dynamics of blackbox ADEA attacks, we now introduce a novel adversarial multi-agent game setting, CyberMARL, that allows to model the temporal co-evolutionary dynamics of both network attackers and defenders explicitly. In CyberMARL (see Figure 7), attackers and defenders perform online training of their adversarial perturbation generators, and NFC networks, respectively. Attackers are rewarded if their generated perturbations evade the current defender policy, while defenders are rewarded if they classify adversarial perturbations correctly. In order to avoid runaway feedback effects, CyberMARL slows the coupling of attacker and defender through the use of a shared adversarial sample buffer. Optionally, we additionally assume that all agents have access to a fixed NFC that has been trained on a ground truth distribution apriori. In each turn, both attacker and defender independently sample a statistical flow feature from the environment: The attacker receives a malicious sample for which she is to generate a suitable adversarial perturbation. In contrast, the defender randomly receives either a malicious, benign or adversarial sample from the environment and needs to correctly classify it as malicious or benign.

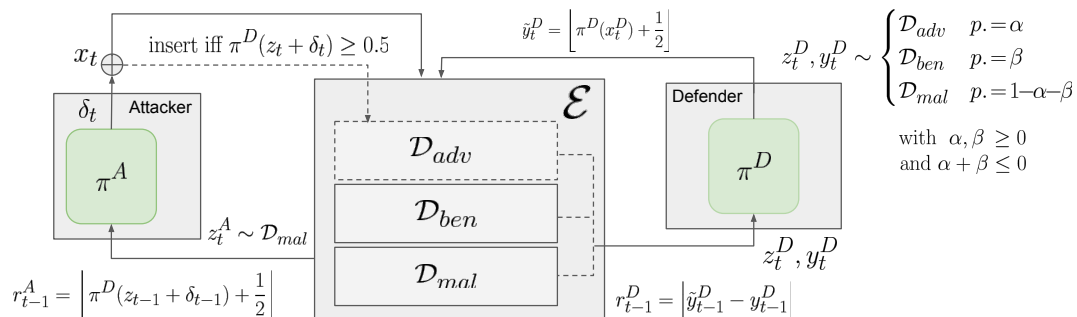


Figure 7: A schematic overview of the CyberMARL setting. Example network architectures for both attackers and defenders may be found in Appendix A.3

In CyberMARL, the environment \mathcal{E} stores all successful adversarial samples generated by the attacker in a buffer \mathcal{D}_{adv} . Otherwise, the attacker follows an identical process as in FAST, with the subtle difference that not C , but

dynamically updating defender policy π^D has to be fooled. CyberMARL also extends FAST by adding a defender agent with policy π^D . At each episode start, the defender receives an i.i.d. sample z^D from the environment which is sampled at probability $\alpha > 0$ from \mathcal{D}_{adv} ⁴, instead from \mathcal{D}_{ben} with probability $\beta > 0$, $\alpha + \beta \leq 1$, and from \mathcal{D} otherwise. The defender then receives a reward of +1 if it classifies z^D correctly into either malicious or benign, i.e. iff $[\pi^D(z^D) + \frac{1}{2}] - y^D = 0$, and 0 otherwise.

Both attacker and defender policies are trained using the MADDPG framework (see section 2.2) under CTDE (see section 2.2). Note that, instead of conditioning on the full state of the environment, we approximate s_t (see section 2.2) by the union of agent observations z_t^A and z_t^D .

Overall, CyberMARL can be summarised as a two-player adversarial game. Importantly, this game is not zero-sum in the conventional sense: rewards for attacker and defender are only weakly coupled through the adversarial sample buffer \mathcal{D}_{adv} , and this coupling takes at least one timestep. This means that the defender policy may have changed in the meantime and an adversarial sample that worked at generation time may happen not to work anymore when sampled by the defender. In addition, CyberMARL can easily be extended with cooperative reward structures that incentivise both attackers and defenders to simulate the interest of both to keep network operations intact.

We let CyberMARL’s policy network take the prediction score of the fixed classifier C as additional input. To ensure that π^D is initialised close to C , we employ a skip connection in π^D ’s network architecture. Each fully connected layer in both actors and critics has 64 activations and we employ the ReLU activation function. We choose $\gamma = 0.95$ and a learning rate of 10^{-2} .

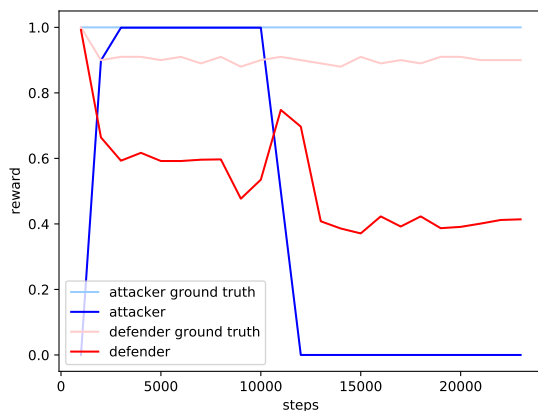


Figure 8: Spontaneous reward transition in CyberMARL framework for $\alpha = 0.1$, $\beta = 0.4$. The defender starts off with considerable skill. The attacker quickly learns how to reliably fool the defender, leading the defender’s skill to drop slightly. A sudden transition renders the attacker entirely unskilled, with the defender temporarily regaining some skill, only to then drop to ca. 0.4. The ground truth baselines indicates what rewards both agents would receive if the defender’s policy was equal to the fixed classifier C . We see that the attacker always perfectly fools C . The evidence suggests that the continuous hardening of π^D triggers a catastrophic transition point for the attacker.

As a case study, we investigate a spontaneous reward transition in the CyberMARL framework for $\alpha = 0.1$, $\beta = 0.4$ (see Figure 8). The defender starts off with considerable skill. The attacker quickly learns how to reliably fool the defender, leading the defender’s skill to drop slightly. A sudden transition renders the attacker entirely unskilled, with the defender temporarily regaining some skill, only to then drop to ca. 0.4. The ground truth baselines indicates what rewards both agents would receive if the defender’s policy was equal to the fixed classifier C . We see that the attacker always perfectly fools C . The evidence suggests that the continuous hardening of π^D triggers a catastrophic transition point for the attacker. However, it remains unclear why the defender stabilises at a reward of around 0.4 specifically. The resulting collapse in defender performance may indicate that the policy network initialisation is unstable. This could be potentially alleviated by replacing the skip connection by a supervised pre-training step.

⁴iff \mathcal{D}_{adv} is not empty, else α is set to effective zero until it has been filled with at least one sample.

5 FIXED POINTS OF ATTACKER-DEFENDER CO-EVOLUTION

5.1 IS THERE EMPIRICAL EVIDENCE OF ATTACKER-DEFENDER CO-EVOLUTION IN LOCKED SHIELDS?

To gain some insight on possible real-world evidence of attacker-defender coevolution, we investigate flow classification in a real-world historic dataset from Locked Shields exercises (see Section 2). While neither strategies of Team Red and Team Blue are publicly known, this paper uses a dataset of $> 100M$ recorded package flow features from the years 2017 to 2019. Using a ground truth of knowingly infiltrated external nodes, this dataset is labeled for C&C flows depending on whether a package flow involved a knowingly infected external node. This labeling process is of course not exact and omits potential package flows that are relayed through infected internal nodes.

To assess the difficulty of separating malicious from benign flows, we train a gradient-boosting random forest classifier (*LightGBM*) (Ke et al., 2017) on a class-balanced dataset of 1m samples, using all 80 features provided. We find that our classifier achieves an accuracy of more than 98% when trained and evaluated on samples from both 2017 and 2018. However, accuracy decreases, and, in particular, false positive rates significantly increase if classifiers trained on a particular year are validated on another year (see Table 4). This implies that not only malicious flows changed slightly between consecutive Locked Shields exercises, but also the characteristics of the benign background flows. As we do not have sufficient information about changes in attacker-defender methodologies over time, however, we cannot establish any causal relationship for these changes, hence cannot definitely ascribe them to co-evolutionary dynamics.

Train \ Val	2017	2018	all
2017	0.9851	0.9680	0.9785
2018	0.9708	0.9956	0.9799
all	0.9799	0.9945	0.9847

Table 2: Predicting malicious network traffic

Train \ Val	2017	2018
2017	0.9(3)	0.5(6)
2018	0.9(5)	0.1(0.8)

Table 3: False negatives (positives), in %

Table 4: ROC under AUC for binary LightGBM classifiers trained and evaluated on pairs of 10^6 class-balanced Locked Shield labels (top 20). Elevated false positive (negative) rates for cross-annual evaluation indicate that distributions of both malicious and benign network traffic changed significantly between exercises.

5.2 WHITELISTING: A POSSIBLE FIXED POINT OF ATTACKER-DEFENDER CO-EVOLUTION?

We argue that the adoption of automated defense systems (AI-NIDS) will stimulate the development of automated attack systems, and in turn dramatically shorten attacker-defender co-evolutionary cycles. This poses the question of whether attacker-defender co-evolution has any natural fixed points, and whether the system will eventually converge to these. We hypothesise that one possible fixed point could be a setting we refer to as *Whitelisting Hell*, in which defenders restrict flow feature space to a dynamic distribution of narrow subspaces, outside of which traffic is dropped automatically. This would counteract the attacker’s capabilities of traversing freely in flow feature space, but limit the networks ad-hoc operability. *Whitelisting Hell* (see Figure 9) therefore poses a fundamental tradeoff between security, and operational flexibility that is already encountered in IoT security research.

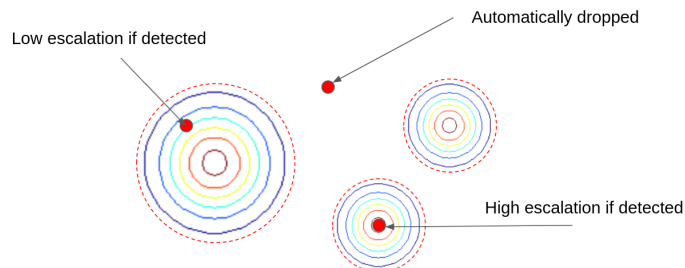


Figure 9: A schematic depiction of *Whitelisting*. The concentric circles denote different whitelisting ‘corridors’, i.e. statistical flow feature subspaces outside of which traffic is automatically blocked. The red dots denote different types of adversarial perturbations. We assume that adversarial perturbations, if falling closer to the centre of each whitelisted corridor, are more likely to cause major system disruption if detected due to the increased amount of background traffic in these areas.

Even in *Whitelisting Hell*, attackers may learn successful ADEA schemes. We suggest that in such cases, one may consider the factorised problem of the attacker first needing to predict which SFF subspaces are whitelisted ahead of time, and then learn to generate adequate adversarial perturbations separately for each subspace. As in CyberMARL, attackers need to ensure that the perturbations generated are unlikely to trigger major security escalations if detected by NIDS infrastructure (particularly behavioural anomaly detection not based on NFC approaches), while keeping botnet operability intact.

5.3 ATTACKER-DEFENDER CO-EVOLUTION: A CONTINUAL LEARNING PERSPECTIVE

Whether in *Whitelisting Hell*, or under the more general ADEA conditions explored by CyberMARL, possible attack strategies may seek to exploit a weakness of supervised classifiers: Neural networks, as well as other classifiers, suffer from catastrophic forgetting. Catastrophic forgetting occurs when a network’s weights do no longer reflect the gradient updates associated with samples that it has not seen in training over an extended period of time. In CyberMARL settings, attackers might exploit defender NFCs by confining itself to perturbation subspaces for a prolonged period of time, before suddenly switching to a different perturbation subspace which the NFC has not been trained on for an extended period of time. In Figure 11, we illustrate various learning instabilities when generative networks are trained on multi-modal data whose modes suddenly change during the learning process.

As an antidote to catastrophic forgetting, training samples could be exhaustively stored and the classifier be retrained continuously, however, this may not be possible given traffic volumes, as well as real-time constraints and computation budgets even of large-scale AI-NIDS systems. We suggest and empirically illustrate that NFC defenses in future AI-NIDS may benefit from *continual learning* (Parisi et al., 2019) techniques. In particular, we suggest that defender networks should employ a reinforcement learning variant of (Chaudhry et al., 2019, A-GEM), which is based on retaining a small set of characteristics samples in a replay buffer that can be used for retraining.

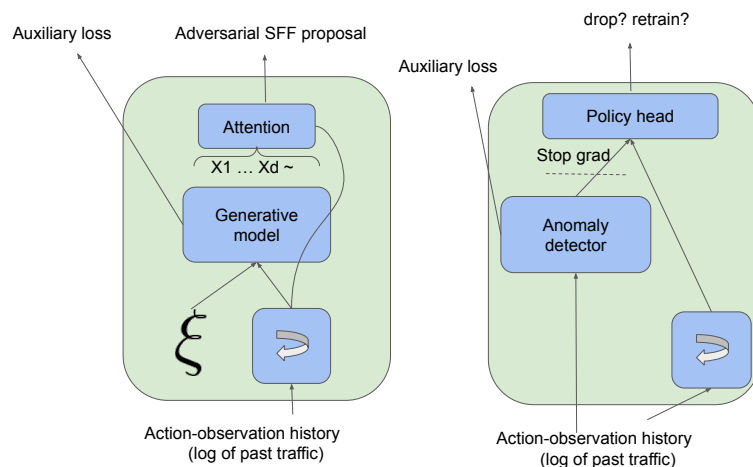


Figure 10: Suggested network architectures for CyberMARL in the continual learning setting. Left: attacker policy network, Right: defender policy network.

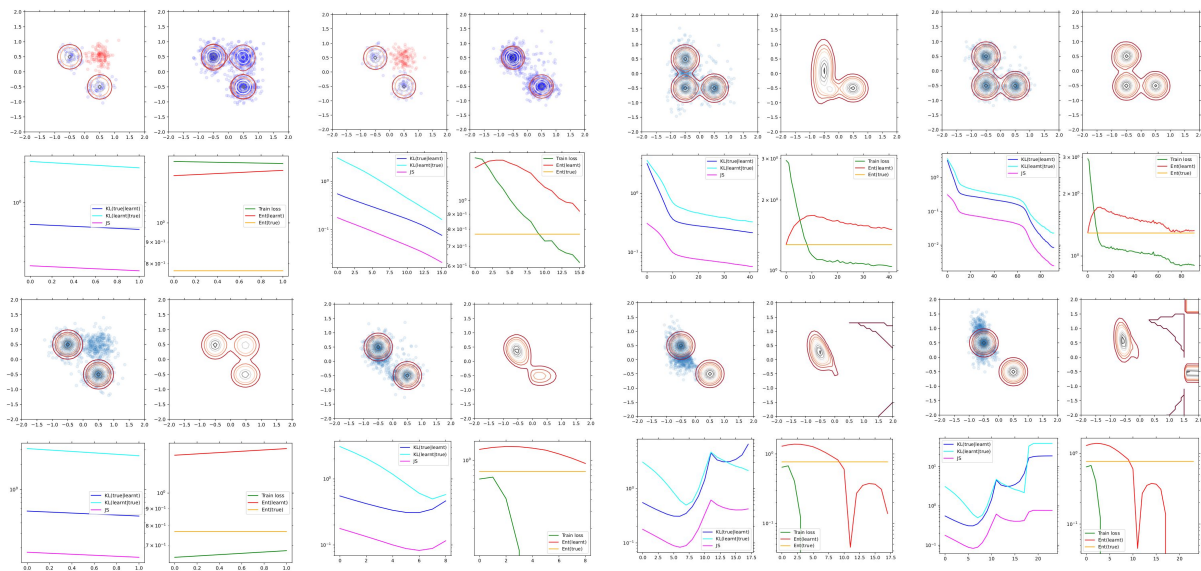


Figure 11: Illustration of the learning process of an autoregressive generative model when unlearning a mode of the distribution (top left and center left), and subsequently adding a distributional mode (top center right and right). Bottom left to right: Unlike at the top, this random initialisation results in an unstable adaption process when sampling distribution has top right mode removed, highlighting the need for a continual learning approach to adversarial perturbation generation in temporally-extended general sum games.

6 RELATED WORK

There exists a large body of literature on *adversarial attacks* against neural network classifiers and we do not guarantee completeness of our literature review⁵. Early methods include the fast gradient sign method (FGSM) (Goodfellow et al., 2015), as well as several optimization-based methods (Carlini & Wagner, 2017; Liu et al., 2017; Xiao et al., 2018; Eykholt et al., 2018). Traditional work on optimal evasion attacks, on and hardening of, non-neural network flow classifiers focuses on the generation of adversarial samples that minimise the L_p -norm to real samples (Merkli, 2020).

Reinforcement learning is a learning setting in which an agent interacts with an environment trying to optimise a scalar reward. Deep reinforcement learning (DRL), which employs DNNs as function approximators, dates back to a seminal paper by (Mnih et al., 2015). Since then, single-agent DRL has found a number of applications in cybersecurity (Nguyen & Reddi, 2020). A number of deep multi-agent reinforcement learning for partially-observable stochastic games have been proposed, ranging from model-free actor-critic variants (OpenAI et al., 2019; Lowe et al., 2020) to Monte-Carlo tree search with self-play (Silver et al., 2017; Schrittwieser et al., 2020).

A number of recent papers study certain cyber security scenarios as temporally-extended adversarial games: Eghtesad et al. (2020) seek to thwart attacks on cyber networks by continuously changing their attack surface, ie. the configuration of hosts and network topology, using deep reinforcement learning. Ye et al. (2021) propose an approach in which a defender adopts differential privacy mechanisms to strategically change the number of systems and obfuscate the configurations of systems, while an attacker adopts a Bayesian inference approach to infer the real configurations of network systems. None of these works explicitly treats botnet infiltration and detection scenarios.

Work concurrent to this paper uses off-policy double Q-learning in order to harden a network flow classifier using adversarial sample generation guided by a sparse reward feedback signal (Apruzzese et al., 2020). Contrary to our framework, they do not consider simultaneous evolution of both attacker and defender. In addition, Apruzzese et al. (2020)’s approach only allows for sparse and *discrete* perturbations to be generated, while our approach based on DDPG allows for arbitrary perturbations to be generated. In particular, the latter crucially allows us to learn perturbations that involve changes to multiple features simultaneously.

⁵If you believe that a particular work needs to be included here, please contact the authors.

7 CONCLUSIONS AND FUTURE WORK

In this whitepaper, we are charting a possible future in which the automation of cyber defense systems (such as AI-NIDS) has accelerated the traditional co-evolutionary cycle between attackers and defenders by orders of magnitude. Consequently, we propose that in such a world, cyber defense research might itself need to depart from finding responses to the latest threats in a reactionary fashion, and instead focus on rigorously studying the co-evolutionary system properties themselves, including their fixed points. At the same time, we have challenged a central tenet in the study of optimal evasion attacks: namely that attackers do not necessarily incur a cost proportional to the size of the adversarial perturbation, but rather can, even today, generate arbitrary ones at negligible cost.

Translating all of these insights to the setting of botnet detection and mitigation, we first introduce FAST, an efficient reinforcement-learning based blackbox adversarial attack generator that can generate both discrete and continuous perturbations of arbitrary size within a single model episode step. We empirically show that FAST can also be efficiently used in conventional minimal evasion distance settings, although a detailed benchmarking against Apruzzese et al. (2020)'s approach is left for future work, as are extensions with probabilistic policies.

With CyberMARL, we introduce a novel setting that allows attacker-defender dynamics to be studied as a temporally-extended general-sum game. While our empirical investigation of this setting are preliminary, CyberMARL opens up a whole new research program for the study of fixed point dynamics of automated attacker-defender systems.

We also hypothesise that a fixed point of attacker-defender co-evolution could be a setting with extensive use of network traffic *whitelisting* and argue that classifiers of future AI-NIDS will require techniques from the continual learning community in order to avoid forced catastrophic forgetting.

ACKNOWLEDGEMENTS

We thank Vincent Lenders, Giorgio Tresoldi, Luca Gambazzi, and Klaudia Krawiecka for helpful discussions. Christian Schroeder de Witt is generously funded by Cyber Defence Campus, armasuisse Science and Technology, Switzerland. This project has received funding from the European Research Council under the European Union's Horizon 2020 research and innovation programme (grant agreement number 637713). The experiments were made possible by a generous equipment grant from NVIDIA.

REFERENCES

- Brendan Abraham, Abhijith Mandya, Rohan Bapat, Fatma Alali, Don E. Brown, and Malathi Veeraraghavan. A Comparison of Machine Learning Approaches to Detect Botnet Traffic. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, July 2018. doi: 10.1109/IJCNN.2018.8489096. ISSN: 2161-4407.
- Moheeb Abu Rajab, Jay Zarfoss, Fabian Monrose, and Andreas Terzis. A multifaceted approach to understanding the botnet phenomenon. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, IMC '06, pp. 41–52, New York, NY, USA, October 2006. Association for Computing Machinery. ISBN 978-1-59593-561-8. doi: 10.1145/1177080.1177086. URL <https://doi.org/10.1145/1177080.1177086>.
- Nik Alexandrov, Dave Cliff, and Charlie Figuero. Exploring Coevolutionary Dynamics of Competitive Arms-Races Between Infinitely Diverse Heterogenous Adaptive Automated Trader-Agents. SSRN Scholarly Paper ID 3901889, Social Science Research Network, Rochester, NY, August 2021. URL <https://papers.ssrn.com/abstract=3901889>.
- G. Apruzzese, M. Andreolini, M. Marchetti, A. Venturi, and M. Colajanni. Deep Reinforcement Adversarial Learning against Botnet Evasion Attacks. *IEEE Transactions on Network and Service Management*, pp. 1–1, 2020. ISSN 1932-4537. doi: 10.1109/TNSM.2020.3031843. Conference Name: IEEE Transactions on Network and Service Management.
- Giovanni Apruzzese, M. Andreolini, Luca Ferretti, Mirco Marchetti, and M. Colajanni. Modeling Realistic Adversarial Attacks against Network Intrusion Detection Systems. *Digital Threats: Research and Practice*, 2021. doi: 10.1145/3469659.
- Marco Barreno, Blaine Nelson, Russell Sears, Anthony D. Joseph, and J. D. Tygar. Can machine learning be secure? In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, ASIACCS '06, pp. 16–25, New York, NY, USA, March 2006. Association for Computing Machinery. ISBN 978-1-59593-272-3. doi: 10.1145/1128817.1128824. URL <https://doi.org/10.1145/1128817.1128824>.

-
- Philippe Biondi and Scapy Community. Scapy, 2021. URL <https://secdev.github.io/>.
- Nicholas Carlini and David Wagner. Towards Evaluating the Robustness of Neural Networks. *arXiv:1608.04644 [cs]*, March 2017. URL <http://arxiv.org/abs/1608.04644>. arXiv: 1608.04644.
- CCDCOE. Locked Shields, 2020. URL <https://ccdcoe.org/exercises/locked-shields/>.
- Arslan Chaudhry, Marc’ Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient Lifelong Learning with A-GEM. *ICLR*, 2019.
- Alesia Chernikova and Alina Oprea. FENCE: Feasible Evasion Attacks on Neural Networks in Constrained Environments. *arXiv:1909.10480 [cs]*, September 2020. URL <http://arxiv.org/abs/1909.10480>. arXiv: 1909.10480.
- Alonzo Church. Application of Recursive Arithmetic to the Problem of Circuit Synthesis. *Journal of Symbolic Logic*, 28(4):289–290, 1963. doi: 10.2307/2271310. Publisher: Association for Symbolic Logic.
- Nilesh Dalvi, Pedro Domingos, Mausam, Sumit Sanghai, and Deepak Verma. Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD ’04, pp. 99–108, New York, NY, USA, August 2004. Association for Computing Machinery. ISBN 978-1-58113-888-7. doi: 10.1145/1014052.1014066. URL <https://doi.org/10.1145/1014052.1014066>.
- Dask Development Team. *Dask: Library for dynamic task scheduling*, 2016. URL <https://dask.org>.
- Jacob Devlin, Jonathan Uesato, Surya Bhupatiraju, Rishabh Singh, Abdel-rahman Mohamed, and Pushmeet Kohli. RobustFill: neural program learning under noisy I/O. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML’17, pp. 990–998, Sydney, NSW, Australia, August 2017. JMLR.org.
- Taha Eghesad, Yevgeniy Vorobeychik, and Aron Laszka. Adversarial Deep Reinforcement Learning based Adaptive Moving Target Defense. *arXiv:1911.11972 [cs]*, August 2020. URL <http://arxiv.org/abs/1911.11972>. arXiv: 1911.11972.
- Kevin Eykholt, Ivan Evtimov, Earlene Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust Physical-World Attacks on Deep Learning Models. *arXiv:1707.08945 [cs]*, April 2018. URL <http://arxiv.org/abs/1707.08945>. arXiv: 1707.08945.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27, pp. 2672–2680. Curran Associates, Inc., 2014. URL <https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf>.
- Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. *arXiv:1412.6572 [cs, stat]*, March 2015. URL <http://arxiv.org/abs/1412.6572>. arXiv: 1412.6572.
- Shixiang Gu, Timothy Lillicrap, Ilya Sutskever, and Sergey Levine. Continuous Deep Q-Learning with Model-based Acceleration. *arXiv:1603.00748 [cs]*, March 2016. URL <http://arxiv.org/abs/1603.00748>. arXiv: 1603.00748.
- Eric A. Hansen, Daniel S. Bernstein, and Shlomo Zilberstein. Dynamic programming for partially observable stochastic games. In *Proceedings of the 19th national conference on Artificial intelligence*, AAAI’04, pp. 709–715, San Jose, California, July 2004. AAAI Press. ISBN 978-0-262-51183-4.
- Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. Inferring and Executing Programs for Visual Reasoning. *arXiv:1705.03633 [cs]*, May 2017. URL <http://arxiv.org/abs/1705.03633>. arXiv: 1705.03633.
- Alex Kantchelian, J. D. Tygar, and Anthony D. Joseph. Evasion and Hardening of Tree Ensemble Classifiers. *arXiv:1509.07892 [cs, stat]*, May 2016. URL <http://arxiv.org/abs/1509.07892>. arXiv: 1509.07892.
- Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: a highly efficient gradient boosting decision tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS’17, pp. 3149–3157, Red Hook, NY, USA, December 2017. Curran Associates Inc. ISBN 978-1-5108-6096-4.

-
- Landon Kraemer and Bikramjit Banerjee. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing*, 190:82–94, 2016.
- Harold Kuhn. Extensive games and the problem of information. *Annals of Mathematics Studies*, 28, 1953.
- Anukool Lakhina, Mark Crovella, and Christophe Diot. Diagnosing network-wide traffic anomalies. *ACM SIGCOMM Computer Communication Review*, 34(4):219–230, August 2004. ISSN 0146-4833. doi: 10.1145/1030194.1015492. URL <https://doi.org/10.1145/1030194.1015492>.
- Arash Habibi Lashkari. ahlashkari/CICFlowMeter, August 2020. URL <https://github.com/ahlashkari/CICFlowMeter>. original-date: 2018-02-12T16:57:30Z.
- Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv:1509.02971 [cs, stat]*, September 2015. URL <http://arxiv.org/abs/1509.02971>. arXiv: 1509.02971.
- Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 8(3-4):293–321, May 1992. ISSN 0885-6125, 1573-0565. doi: 10.1007/BF00992699. URL <https://link.springer.com/article/10.1007/BF00992699>.
- Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into Transferable Adversarial Examples and Black-box Attacks. *arXiv:1611.02770 [cs]*, February 2017. URL <http://arxiv.org/abs/1611.02770>. arXiv: 1611.02770.
- Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. *arXiv:1706.02275 [cs]*, June 2017. URL <http://arxiv.org/abs/1706.02275>. arXiv: 1706.02275.
- Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. *arXiv:1706.02275 [cs]*, March 2020. URL <http://arxiv.org/abs/1706.02275>. arXiv: 1706.02275.
- Roland Meier, Artūrs Lavrenovs, Kimmo Heinäaro, Luca Gambazzi, and Vincent Lenders. Towards an AI-powered Player in Cyber Defence Exercises. In *2021 13th International Conference on Cyber Conflict (CyCon)*, pp. 309–326, May 2021. doi: 10.23919/CyCon51939.2021.9467801. ISSN: 2325-5374.
- Yannick Merkli. *Evaluating and Defeating Network Flow Classifiers Through Adversarial Machine Learning*. PhD thesis, ETH Zurich, 2020.
- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharmashan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015. ISSN 1476-4687. doi: 10.1038/nature14236. URL <https://www.nature.com/articles/nature14236>. Number: 7540 Publisher: Nature Publishing Group.
- Blaine Nelson, Benjamin Rubinstein, Ling Huang, Anthony Joseph, Shing-hon Lau, Steven Lee, Satish Rao, Anthony Tran, and Doug Tygar. Near-Optimal Evasion of Convex-Inducing Classifiers. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 549–556. JMLR Workshop and Conference Proceedings, March 2010. URL <http://proceedings.mlr.press/v9/nelson10a.html>. ISSN: 1938-7228.
- Thanh Thi Nguyen and Vijay Janapa Reddi. Deep Reinforcement Learning for Cyber Security. *arXiv:1906.05799 [cs, stat]*, July 2020. URL <http://arxiv.org/abs/1906.05799>. arXiv: 1906.05799.
- Frans A. Oliehoek and Christopher Amato. *A Concise Introduction to Decentralized POMDPs*. SpringerBriefs in Intelligent Systems. Springer International Publishing, 2016. ISBN 978-3-319-28927-4. doi: 10.1007/978-3-319-28929-8. URL <https://www.springer.com/gp/book/9783319289274>.
- OpenAI, Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique Pondé de Oliveira Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. Dota 2 with Large Scale Deep Reinforcement Learning. *arXiv:1912.06680 [cs, stat]*, December 2019. URL <http://arxiv.org/abs/1912.06680>. arXiv: 1912.06680.

-
- German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, May 2019. ISSN 0893-6080. doi: 10.1016/j.neunet.2019.01.012. URL <https://www.sciencedirect.com/science/article/pii/S0893608019300231>.
- Emilio Parisotto, Abdel-rahman Mohamed, Rishabh Singh, Lihong Li, Dengyong Zhou, and Pushmeet Kohli. Neuro-Symbolic Program Synthesis. *arXiv:1611.01855 [cs]*, November 2016. URL <http://arxiv.org/abs/1611.01855>. arXiv: 1611.01855.
- Kui Ren, Tianhang Zheng, Zhan Qin, and Xue Liu. Adversarial Attacks and Defenses in Deep Learning. *Engineering*, 6(3):346–360, March 2020. ISSN 2095-8099. doi: 10.1016/j.eng.2019.12.012. URL <http://www.sciencedirect.com/science/article/pii/S209580991930503X>.
- Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model. *arXiv:1911.08265 [cs, stat]*, February 2020. URL <http://arxiv.org/abs/1911.08265>. arXiv: 1911.08265.
- David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dhharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *arXiv:1712.01815 [cs]*, December 2017. URL <http://arxiv.org/abs/1712.01815>. arXiv: 1712.01815.
- Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning, second edition: An Introduction*. Bradford Books, Cambridge, Massachusetts, second edition edition edition, November 2018. ISBN 978-0-262-03924-6.
- Ming Tan. Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents. In *In Proceedings of the Tenth International Conference on Machine Learning*, pp. 330–337. Morgan Kaufmann, 1993.
- David Wagner and Paolo Soto. Mimicry attacks on host-based intrusion detection systems. In *Proceedings of the 9th ACM conference on Computer and communications security, CCS '02*, pp. 255–264, New York, NY, USA, November 2002. Association for Computing Machinery. ISBN 978-1-58113-612-8. doi: 10.1145/586110.586145. URL <https://doi.org/10.1145/586110.586145>.
- Richard J. Waldinger and Richard C. T. Lee. PROW: a step toward automatic program writing. In *Proceedings of the 1st international joint conference on Artificial intelligence, IJCAI'69*, pp. 241–252, San Francisco, CA, USA, May 1969. Morgan Kaufmann Publishers Inc.
- Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially Transformed Adversarial Examples. *arXiv:1801.02612 [cs, stat]*, January 2018. URL <http://arxiv.org/abs/1801.02612>. arXiv: 1801.02612.
- D. Ye, T. Zhu, S. Shen, and W. Zhou. A Differentially Private Game Theoretic Approach for Deceiving Cyber Adversaries. *IEEE Transactions on Information Forensics and Security*, 16:569–584, 2021. ISSN 1556-6021. doi: 10.1109/TIFS.2020.3016842. Conference Name: IEEE Transactions on Information Forensics and Security.

A APPENDIX

A.1 DATA REWRITING

The large number of samples contained in the Locked Shields dataset and resulting CSV-file sizes (>100GB) are prohibitively slow to access, even using distributed frameworks such as Dask (Dask Development Team, 2016), for both exploratory analysis, as well as for model training. In contrast to storage formats requiring *read* system calls, including HDF5⁶, Zarr⁷ or xarray⁸, memory-mapped files use the *mmap* system call to map physical disk space directly to virtual process memory, enabling the use of *lazy* OS demand paging and circumventing the kernel buffer. This makes memmaps particularly efficient for random access patterns, such as commonly found during model training.

We therefore rewrite each data feature into an individual *numpy memmap* file. As we choose the primary axis to be ordered by time, access to specific time indices can be efficiently handled using *numpy's searchsorted*.

Feature ID	Mean0	Mean1	Std0	Std1
Protocol	12.4	6.0	5.51	9.6E-2
dstIntExt	0.13	0.10	0.34	0.05
Active Mean	2E6	1E6	3E4	3E5
Init Fwd Win Byts	3E9	8E5	2E9	6E7
FIN Flag Cnt	0.18	0.97	0.38	0.17
Bwd Pkt Len Min	34.7	0.0	101.2	0.2
Flow Pkts/s	2E4	2E3	1E5	4E4
Fwd IAT Max	1E6	8E4	2E6	6E5
Fwd IAT Min	2E5	2E4	1E6	2E5
Subflow Fwd Pkts	8.6	11.5	141.2	7.4
Flow IAT Max	1.5E6	8E5	2.7E6	6E5
Fwd IAT Tot	2E6	1E5	3E6	9E5
Subflow Bwd Pkts	3.4	6.0	196.0	7.0
Subflow Fwd Byts	3.83E	1.5E3	1E5	7.6E3
Bwd Header Len	63	131	4.4E3	141
Tot Bwd Pkts	3.4	6.0	196	7
Fwd Pkt Len Std	26	197	81	44
Fwd Seg Size Min	15.6	20.3	10.1	1.6
Bwd Pkt Len Std	20	86	74	36
Bwd IAT Mean	1E5	1E4	5E5	1E5

Table 5: Two-column list of the 20 statistical flow features (CICFlowMeter) used across all empirical evaluations in this paper. Means and standard deviations are listed for both benign (0) and malicious (0) features. Note that even large differences in dataset statistics do not necessarily imply that individual samples are easily distinguished. Features were derived using feature importance scores after training on all 80 features available using a LightGBM model.

⁶[https://portal.hdfgroup.org/display/HDF5/HDF5\(2021\)](https://portal.hdfgroup.org/display/HDF5/HDF5(2021))

⁷<https://zarr.readthedocs.io/en/stable/> (2021)

⁸<http://xarray.pydata.org/en/stable/> (2021)

A.2 RL TRAINING PATHOLOGIES

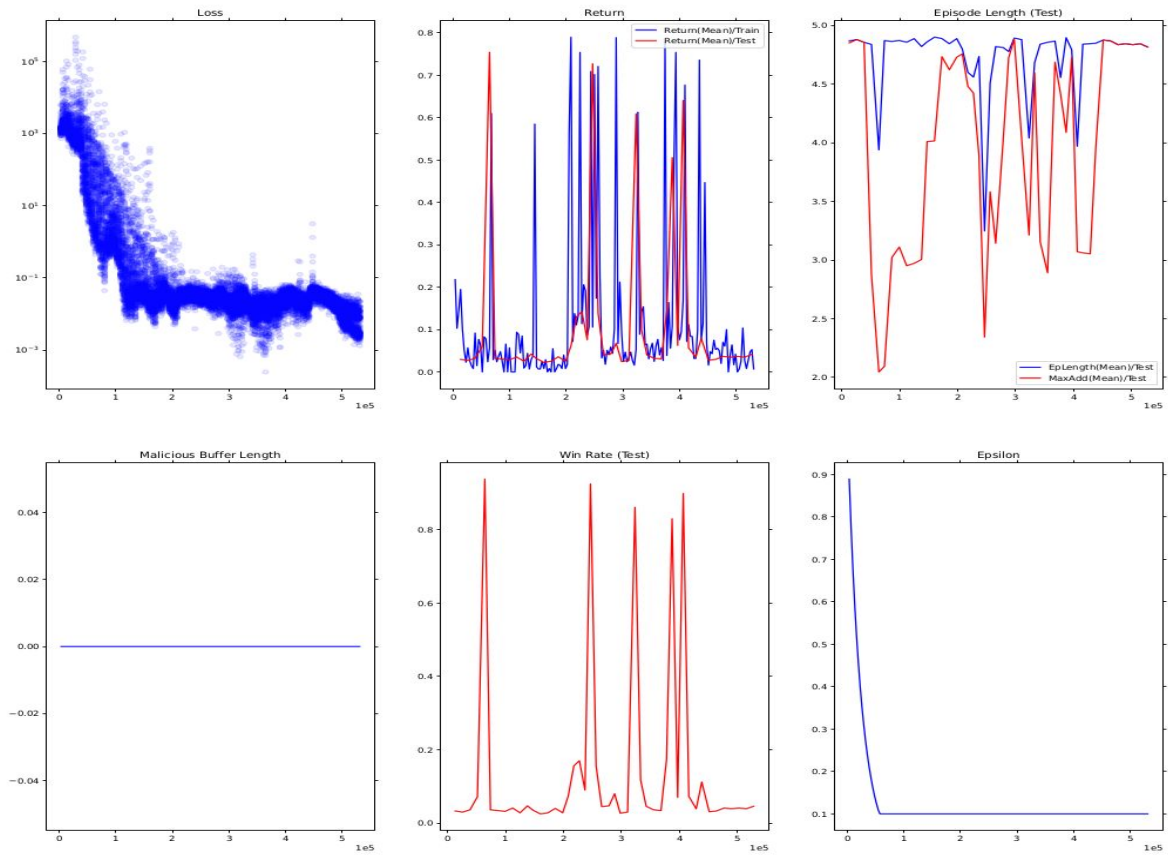


Figure 12: An illustration of common learning pathologies in RL-based blackbox attack training (here within the framework of Apruzzese et al. (2020)): We erroneously chose not to condition our agents' observations on either timesteps (in cases with maximum timestep cutoffs) or the current perturbation delta (for cases in which a episodes are terminated once a maximum feature space distance to the original has been reached).

A.3 EXAMPLE POLICY NETWORK ARCHITECTURES

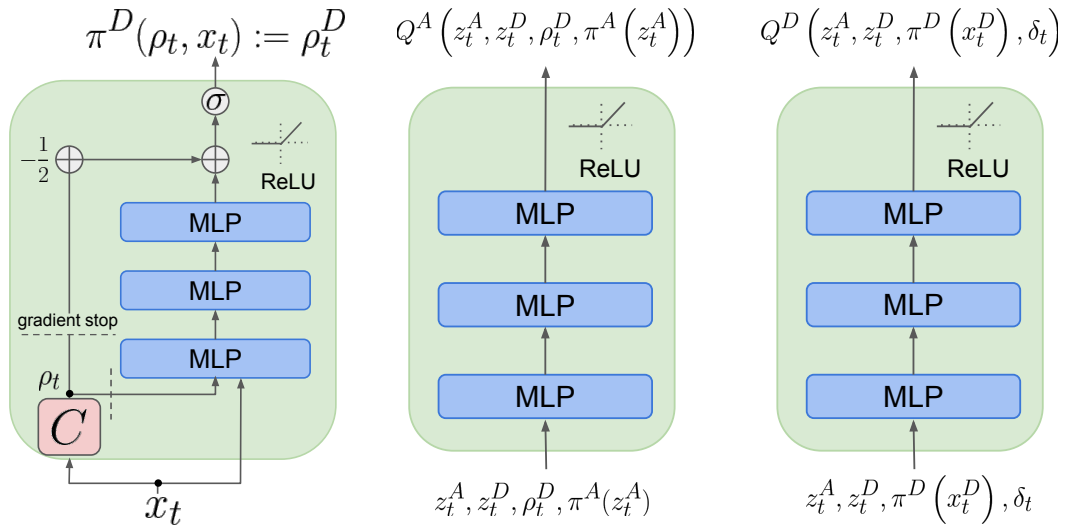


Figure 13: Defender policy network architecture

Figure 14: Critic network architectures for both attacker and defender (CyberMARL)

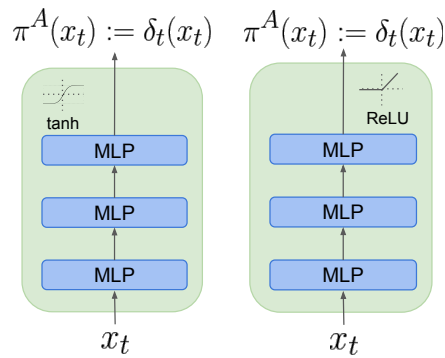


Figure 15:
(FAST)
Attacker policy
network architecture

Figure 16:
(CyberMARL)
Attacker policy
network architecture