Adam-based Augmented Random Search for Control Policies for Distributed Energy Resource Cyber Attack Mitigation

Daniel Arnold[†,⋆], Sy-Toan Ngo[†,⋆], Ciaran Roberts[†], Yize Chen[†], Anna Scaglione[◦], Sean Peisert[†]
[†]*Lawrence Berkeley National Laboratory* {dbarnold,sytoanngo,cmroberts,ychen,sppeisert}@lbl.gov
[◦]*Cornell Tech* as337@cornell.edu
[⋆]*These authors contributed equally to this effort*

***Abstract*— Volt-VAR and Volt-Watt control functions are mechanisms that are included in distributed energy resource (DER) power electronic inverters to mitigate excessively high or low voltages in distribution systems. In the event that a subset of DER have had their Volt-VAR and Volt-Watt settings compromised as part of a cyber-attack, we propose a mechanism to control the remaining set of non-compromised DER to ameliorate large oscillations in system voltages and large voltage imbalances in real time. To do so, we construct control policies for individual non-compromised DER, directly searching the policy space using an Adam-based augmented random search (ARS). In this paper we show that, compared to previous efforts aimed at training policies for DER cybersecurity using deep reinforcement learning (DRL), the proposed approach is able to learn optimal (and sometimes linear) policies an order of magnitude faster than conventional DRL techniques (e.g., Proximal Policy Optimization).**

I. Introduction

Distributed Energy Resources (DER), most notably rooftop solar photovoltaic (PV) systems, are envisioned to be key components of realizing local, state, and federal renewable energy goals in the next several decades. On the federal level, discussions of a national clean energy standard [1] and efforts to further bring down the cost of solar [2] will undoubtedly serve to accelerate already high levels of adoption of rooftop solar PV systems. Faced with a future grid where a large amount of power is generated from DER, stakeholders have convened over the past two decades to codify standards that seek to govern the behavior of DER to ensure resiliency and reliability of the power system.

A standard that has garnered significant attention in recent years is IEEE 1547 [3]. Among other things, this standard proposes mechanisms that allow DER smart inverters to adjust their active and reactive power outputs in response to changes in local system voltages. In so doing, the standard seeks to ensure that DER will collectively act to minimize occurrences of over- or under-voltages.

Two mechanisms proposed within IEEE 1547 aimed at providing inverter-based voltage regulation are known as Volt-VAR (VV) and Volt-Watt (VW) control functions. These control functions, depicted in Figs. 2 - 3, alter DER reactive and active power injection according to piecewise linear non-increasing functions of the locally sensed voltage. In addition to prescribing the general structure of the control functions, IEEE 1547 also specifies different parameterizations of the piecewise linear curves, which are realized via adjusting the parameters $\boldsymbol{\eta} = [\eta_1, \eta_2, \eta_3, \eta_4, \eta_5]$. The ability to adjust the parameters of the VV/VW curves, in theory, allows the dynamic response of DER to changing system voltages to be tailored to specific networks, or regions within a given network.

While the flexibility offered by IEEE 1547 ensures that DER dynamic voltage response can be adjusted to better serve different types of networks (e.g., radial vs. meshed systems), the ability to adjust DER dynamics coupled with Internet, cellular, or power line carrier connectivity of the inverters themselves, potentially allows remote updating of VV/VW parameters by an entity with malicious intention [4]. An excellent example of the extent to which aggregations of smart inverters can be remotely updated was illustrated in Hawaii, where local utilities worked with a smart inverter vendor to remotely update the autonomous control functions of 800,000 inverters in a single day [5].

In the event that a portion of the DER smart inverter functions in a given network have had their settings adjusted as part of a cyber-attack, our previous works (Roberts et. al. [6], [7]) explored the use of Deep Reinforcement Learning (DRL) to determine optimal control policies that alter the behavior of the remaining population of *non-compromised* DER to attempt to mitigate the effect of the cyber-attack in real time. In our first paper, we utilized Proximal Policy Optimization (PPO) to train optimal polices that adjust DER smart inverter VV and VW functions to mitigate attacks aimed at creating large oscillations in system voltages [6]. In a subsequent paper, we extended this framework using a different reward function to develop optimal policies for mitigating attacks designed to create large voltage imbalances in multi-phase systems [7].

The PPO-based policies developed in our previous efforts proved successful in mitigating large oscillations and voltage imbalances even when almost half of the PV smart inverter control functions in a given network were compromised during the cyber-attack. The main drawback of the DRL architecture used to train policies to manage non-compromised DER was lengthy training time. Training on relatively small

This research was supported in part by the Director, Cybersecurity, Energy Security, and Emergency Response, Cybersecurity for Energy Delivery Systems program, of the U.S. Department of Energy, under contract DE-AC02-05CH11231. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors of this work.

networks, such as the IEEE 37-node test feeder using modest resources (Intel® Core™ i7-8850H CPU @ 2.60GHz, 16GB RAM) took several hours. Training control policies on the IEEE 8500 node test feeder would often take tens of hours. While proper parallelization and HPC/cloud resources could be leveraged to bring down the overall training time and enable these solutions to scale to larger networks, many utilities (rural electric cooperatives in the U.S., for example) lack the expertise and financial resources to properly engage these assets for training unique policies for different networks. In order to democratize the training of network-specific control policies that optimally manage DER to ensure grid stability in the face of cyber attacks, additional effort is needed to develop alternative optimal control strategies which can train equivalent policies in substantially less time.

In contrast to many popular reinforcement learning techniques, evolutionary strategies (ES) do not attempt to approximate both the value function and the optimal policy of a Markov decision process (MDP). ES, rather, directly search for optimal policies through perturbing the present policy and evaluating the improvement in the reward by conducting "rollouts" of the MDP under the perturbed policy. Variants of this "perturb and observe" paradigm include genetic algorithms (GAs) from stochastic optimization [8] as well as extremum seeking techniques from nonlinear control theory [9]. In a compelling piece, Salimans et. al. [10] demonstrated that random search (an extremely simple type of evolutionary strategy) can be leveraged to learn competitive and highly scalable policies compared to popular RL techniques for several MDPs. Extending this work, Mania et. al. [11] proposed the *augmented random search* (ARS) algorithm that was used to train *linear* policies for several challenging MDP learning tasks with excellent sample efficiency.

In this paper, motivated by the success of ARS in learning optimal policies for complex MDPs with improved sample efficiency, we consider an extension of ARS with an adaptive learning rate applied to learn optimal policies for DER cyber-resiliency. By incorporating the *Adam optimization method* [12] into the gradient update step of ARS, we are able to train *linear* policies to mitigate voltage oscillations and neural-network based policies to mitigate voltage imbalances caused by cyber-attacked DER. Our experiments show that the Adam-based ARS approach (henceforth referred to as Adam-ARS) enables training of competitive policies an order of magnitude faster than PPO and with less variance than ARS. As such, this work stands to enable efficient and scalable learning of optimal policies for controlling DER smart inverters on a network-specific basis.

This paper is organized as follows. Dynamic models of the DER smart inverter VV and VW control functions, as well as descriptions of the Adam solver and ARS are discussed in Section II. Section III presents the Adam-ARS algorithm and discusses the framework for training optimal policies. Results showing the effectiveness of the trained policies in mitigating voltage oscillations and imbalances introduced by cyber-attacked DER are provided in Section IV. Finally, concluding remarks are provided in Section V.

II. Preliminaries

A. Inverter Dynamic Modeling and Agent Interaction

The goal of this work is to train an intelligent agent capable of adjusting the parameters of VV/VW control functions in *non-compromised* DER to mitigate large voltage oscillations and imbalances introduced by subsets of DER which have been cyber-attacked. A depiction of the agent interacting with a dynamic model of a DER smart inverter is depicted in Fig. 1.

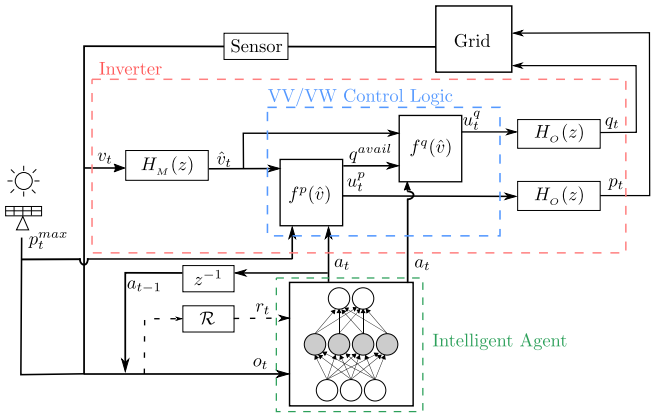

Fig. 1: Block diagram of VV and VW control logic of an inverter.

As is shown in the figure, the smart inverter dynamic model maps grid voltages v_t into active p_t and reactive q_t power injections. The grid voltage is first passed through a low-pass filter $H_M(z)$ to generate the measured grid voltage $\hat{v}_t$ which is fed into the nonlinear Volt-VAR and Volt-Watt control functions, represented by $f^q(\hat{v})$ and $f^p(\hat{v})$. These functions compute reactive and active power setpoints u_t^q and u_t^p which are passed through low pass filters $H_o(z)$ to create the actual powers injected into the grid, q_t and p_t. The VW/VV control functions are shown in Figs. 2 - 3 and consist of piecewise linear non-increasing functions of grid voltage parameterized by a set of voltage values $\boldsymbol{\eta} = [\eta_1, \eta_2, \eta_3, \eta_4, \eta_5]$. The VV/VW control structure depicted in Fig. 1 is known as Volt-Watt precedence [13] where priority is given to the VW controller to compute the active power setpoint u_t^p. Following the computation of u_t^p, any additional inverter capacity can be used for reactive power generation, q_{avail}, which is input into the VV controller to determine the setpoint u_t^q. We note that this dynamic model of smart inverter behavior is consistent with IEEE 1547 and supporting documents [3], [13].

The intelligent agent interacts directly with the VV/VW controllers through an action taken at time t (a_t). The subsequent reward r_t due to the application of a_t is then input into the agent, along with a set of observations o_t from the grid as well as past actions a_{t-1}.

B. Observations of Voltage Oscillations and Imbalances

During a cyber-attack, we assume that an adversary has the capability to alter the parameter vector $\boldsymbol{\eta}$ in a portion of DER thereby adjusting the shape of their VV/VW curves

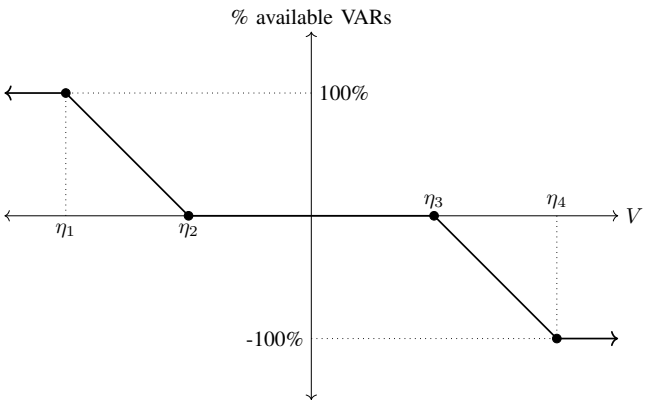

Fig. 2: Inverter Volt-VAR curve. Positive values denote VAR injection.

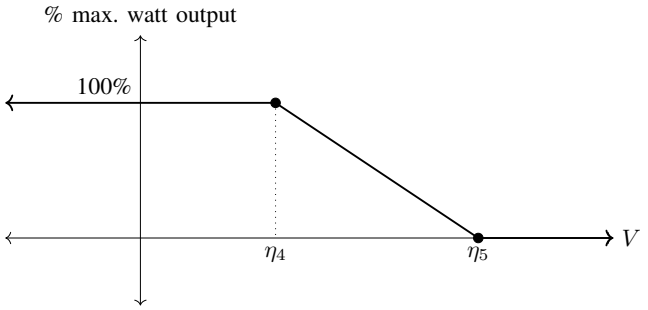

Fig. 3: Inverter Volt-Watt curve. Positive values denote watt injection.

to generate large voltage oscillations [6] or large voltage imbalances [7]. We then assume that the remaining non-compromised DER in the system are equipped with an intelligent agent depicted in Fig. 1 which will re-adjust the parameter vector $\boldsymbol{\eta}$ to mitigate the cyber-attack in real time.

Critical observations needed by the agent for cyber-attack mitigation are measurements of the intensity of the voltage oscillations and imbalances computed from grid telemetry. We refer the reader to our previous works [6], [7] for detailed discussions of how these observations are computed in real time. The procedure is briefly summarized here:

Voltage Oscillations (VO): We utilize an intuitive filtering process to extract the "energy" associated with observed voltage oscillations. The filter consists of the series connection of a high-pass filter H_{HP}, a signal square element (with positive gain c), and a low-pass filter H_{LP}, shown in Fig. 4. The output of the filter $\mathbf{vo}_{i,t}$ is a non-negative value which becomes larger as the amplitude of the oscillations in node i voltage ($v_{i,t}$) increase. For proper operation, the high and low-pass filter critical frequencies should be chosen as to not attenuate oscillations resulting from cyber-attacked inverters.

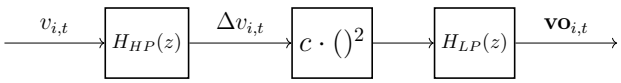

Fig. 4: Block diagram of illustrating the filtering process used to compute a measurement of the intensity of voltage oscillations.

Voltage Imbalances (VI): The metric used to compute voltage imbalance at node i at time t is given by:

$$\mathrm{vi}_{i,t} = \frac{\max(|\bar{v}_{i,t} - v^a_{i,t}|, |\bar{v}_{i,t} - v^b_{i,t}|, |\bar{v}_{i,t} - v^c_{i,t}|)}{\bar{v}_{i,t}} \quad (1)$$

where $\bar{v}_i$ denotes the mean measured voltage magnitude at bus i, and $v^a_{i,t}$, $v^b_{i,t}$, $v^c_{i,t}$ are the measured voltage magnitudes on phase a, b, and c respectively.

C. Adam

Adam is a first-order gradient-based stochastic optimization method that which utilizes adaptive estimates of lower-order moments [14]. Similar to AdaGrad [12] and RMSProp [15], the method features an adaptive learning rate and has been shown to work well with sparse gradients and in non-stationary settings. We present the Adam algorithm for completeness in Algorithm 1.

D. Random Search and Augmented Random Search

The goal of reinforcement learning is to determine a policy that governs a dynamic system to maximize a reward associated with a specific task. This problem can be formulated as [11]:

$$\max_{\theta \in \mathbb{R}^n} \mathbb{E}_\xi[r(\pi_\theta, \xi)], \quad (2)$$

where π_θ is the policy parameterized by $\theta \in \mathbb{R}^n$, ξ represents the randomness of the environment, $r(\pi_\theta, \xi)$ is the cumulative reward that the policy π_θ generates in one trajectory (i.e., a "rollout") of the system.

In situations where the gradient of the objective function is not directly available to aid in the search for for the optimal parameter vector θ^*, one can utilize *measurements* of the reward to approximate the gradient of the reward under the present policy $g(\theta)$. A *finite-difference* approximation of $g(\theta)$ is given by [8]:

$$g(\theta) = \frac{r(\pi_{\theta+c\delta}, \xi_1) - r(\pi_{\theta-c\delta}, \xi_2)}{2c}, \quad (3)$$

where c is a positive scalar and δ is a zero-mean and Gaussian vector. Random Search (RS) [8], [10], [11] incorporates single or mini-batches of gradient approximations of (3) into simple gradient ascent to search for θ^*.

Mania et. al. [11] suggest several improvements to RS, yielding the Augmented Random Search (ARS) algorithm. The modifications are:

- Normalization of the states
- Scaling the gradient by the standard deviation of return
- Using top performing directions in mini-batch updates

III. Methodology

A. Adam-based Augmented Random Search

We propose an extension of ARS based on the replacement of vanilla gradient ascent with the Adam optimization algorithm. The method is specified in Algorithm 2.

Algorithm 1: Adam Optimization Algorithm 1-step forward

Hyperparameters: Gradient g_t, stepsize α, exponential decay rate β_0, β_1 for moment estimates, tolerance parameter $\lambda_{ADAM} > 0$ for numerical stability. $m_0, v_0 \leftarrow [0, 0, 0]$

Function ADAM$(\theta_j, g_j, \alpha, \beta_0, \beta_1)$**:**

$m_j \leftarrow \beta_1 \cdot m_{j-1} + (1 - \beta_1) \cdot g_j$

$v_j \leftarrow \beta_2 \cdot v_{j-1} + (1 - \beta_1) \cdot g_j^2$

$\hat{m_j} \leftarrow m_j / (1 - \beta_1^j)$

$\hat{v_j} \leftarrow v_j / (1 - \beta_2^j)$

$\theta_{j+1} \leftarrow \theta_j - \alpha \cdot \hat{m_j} / (\sqrt{\hat{v_j}} + \lambda_{ADAM})$

return θ_t

Algorithm 2: ADAM-based Augmented Random Search

Hyperparameters: number of directions sampled per iteration N, exploration noise ν, number of top-performing directions b $(b \leq N)$

Initialize: $\mu_0 = \mathbf{0} \in \mathbb{R}^n, \Sigma_0 = \mathbf{I}_n \in \mathbb{R}^{n \times n}, j = 0$

$$\pi_\theta = \begin{cases} \mathbf{0} \in \mathbb{R}^{p \times n} \text{ if using linear policy} \\ \mathrm{NN}(\mathbf{0}) \text{ if using non-linear policy} \end{cases}$$

while *ending condition not satisfied* **do**

Sample $\delta_1, \delta_2, ..., \delta_N$ of appropriate dimension with i.i.d. standard normal entries.

Collect $2N$ rollouts of horizon H and their corresponding rewards using the $2N$ policies.

$$\pi_{\theta_j,k,+}(\tilde{x}) = \pi_{\theta_j + \nu\delta_k}(\tilde{x}) \quad \text{and} \quad \pi_{\theta_j,k,-}(\tilde{x}) = \pi_{\theta_j - \nu\delta_k}(\tilde{x})$$
$$\text{where } \tilde{x} = \mathrm{diag}(\Sigma_j)^{\frac{-1}{2}}(x - \mu_j)$$

for $k \in \{1, 2, 3, ...N\}$.

Sort the direction δ_k by $\max\{r(\pi_{\theta_j,k,+}), r(\pi_{\theta_j,k,-})\}$, denote by $\delta_{(k)}$ the k-th largest direction, and by $\pi_{\theta_j,(k),+}$ and $\pi_{\theta_j,(k),-}$ the corresponding policies.

Policy update step:

$$g_j = \frac{\alpha}{b\sigma_R} \sum_{k=1}^{b} [r(\pi_{\theta_j,(k),+}) - r(\pi_{\theta_j,(k),-})]\delta_{(k)}$$
$$\theta_{j+1} = \mathrm{ADAM}(\theta_j, g_j, \alpha, \beta_0, \beta_1)$$

where σ_R is the standard deviation of the $2b$ directions used to update step.

Set μ_{j+1}, Σ_{j+1} to be mean and the covariance of the $2NH(j+1)$ states encountered from the start of training.

$j \leftarrow j + 1$

end

B. Training Framework

We seek to train a policy to mitigate large voltage oscillations (VO) and voltage imbalances (VI) stemming from DER smart inverter VV/VW control functions with maliciously chosen setpoints. We represent a single distribution feeder as a graph $G = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ and $\mathcal{L}$ denote the set of nodes and lines, respectively. For simplicity of presentation, it is assumed that a DER equipped with VV/VW functionality is located at every node in $\mathcal{G}$. We now partition the set of inverters into two groups $\mathcal{H}$ and $\mathcal{U}$ representing "compromised" and "non-compromised" sets of DER, where $\mathcal{H} \bigcup \mathcal{U} = \mathcal{N}$. We also make the assumption that the set $\mathcal{U}$ is nonempty and contains sufficient amounts of DER to mitigate the effect of the cyber-attack. We adopt the following framework for training optimal policies:

Training:

1) We define a single agent whose input observation vector is from a single node in the network at time t with worst case VU and VO. The agent has a multi-head output action, $a_t^i \forall i \in \{a, b, c\}$, which is a deviation/offset, $\Delta\eta$, from default VV/VW control curves in Figs. 2 - 3 that is applied across all single-phase inverters. An example of an action is shown in Fig. 5. The action space is a continuous offset between -0.1 pu to 0.1 pu around the default inverter VV/VW curve.
2) In order to preserve the Markov property, new parameterizations of VV/VW functions occurs on a slower timescale than the filter dynamics in Fig. 1.

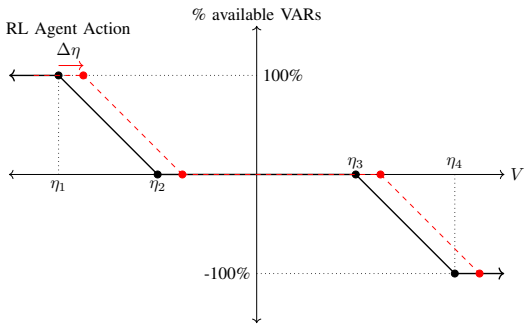

Fig. 5: Action example.

Observation: The observation vectors $o_{i,t}$, where $i \in \mathcal{U}$, at each controller consist of:

1) vi_t: defined in (1).
2) vo_t: defined in Fig. 4.
3) $q_t^{\text{avail, nom}}$: the available reactive power capacity without active power curtailment.
4) a_{t-1}^{a}, a_{t-1}^{b}, a_{t-1}^{c}: the previous action taken by the agent across each phase.
5) v_t^{a}, v_t^{b}, v_t^{c}: voltage phase measurements at time t

During the training of the agent, vo_t and vu_t are the *worst-case* oscillations and imbalances in the feeder, and $q_t^{\text{avail, nom}}$ is the average across all non-compromised inverters. When the agent is deployed to manage individual DER, vo_t and vu_t will be collected from the nearest three-phase node.

Reward: At a time t, the reward function, $r_t(a_t, o_t)$ for a single agent in training is:

$$r_t = -\left(\sigma_u||\text{vi}_t||_\infty + \sigma_u||\text{vo}_t||_\infty + \sum_{i\in\{a,b,c\}} \sigma_a \mathbf{1}_{a_t^i \neq a_{t-1}^i} + \sum_{i\in\{a,b,c\}} \sigma_0 \|a_t^i\|_2 + \frac{1}{|\mathcal{U}|}\sum_{j=1}^{|\mathcal{U}|} \sigma_p \left(1 - \frac{p_{j,t}}{p_{j,t}^{\max}}\right)^2\right). \tag{4}$$

The first term seeks to minimize the maximum imbalance, $||\text{vi}_t||_\infty$ and the second term seeks to minimize the maximum oscillation, $||\text{vu}_t||_\infty$ over all nodes in the network. For details on the remaining components of the reward, we refer the reader to our previous work [7].

IV. Results

A. Experimental Setup

Experiments were conducted on the IEEE-37 node test feeder where DER with VV/VW capability were placed at all load buses with peak active power generation of 100% of nominal load. Additionally, the inverter capacity associated with all DER was oversized by 10% to allow for some reactive power compensation without active power curtailment at all simulation timesteps. Agent training occurs using 700 second rollouts with 1 second timestep simulations using OpenDSS. Load, solar generation, percentage of DER resource which are compromised, and the phase of the voltage regulator are all randomized for each rollout. At $t = 200$ seconds in the simulation, a simulated attack is launched where the attacker controls between 10% and 40% of inverter capacity in the system.

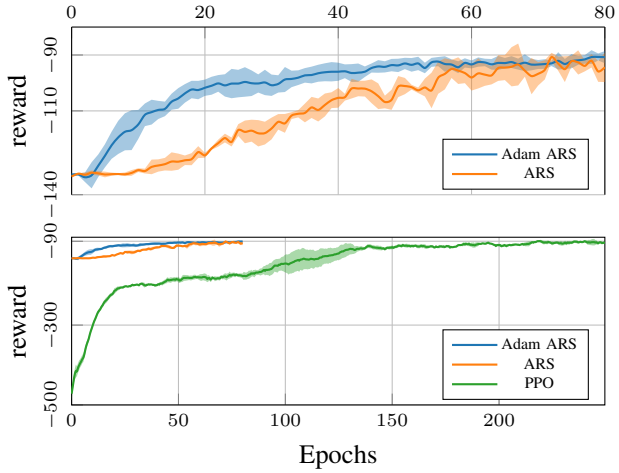

Fig. 6: Average training reward. The shaded area represents the standard deviation over 10 runs.

B. Training performance

Fig. 6 shows the training performance of Adam-ARS, ARS, and PPO when these agents are learning to mitigate a voltage imbalance attack. All three algorithms utilize the same neural network architecture. As is shown in the figures, ARS and ADAM-based ARS take approximately 80 epochs to converge to the optimal policy while PPO takes more than 3 times the number of epochs to converge. Interestingly, due to the nature of training a deep neural network with backpropagation [16], the weights of neural network in PPO requires fine-tuned initialization. This initialization results in extremely poor performance of PPO at the start of training, taking over 100 epochs to achieve rewards similar to both ARS variants. In contrast, since ARS/Adam-ARS randomly perturbs the parameters of the neural network, neural networks for these algorithms can be initialized with weights equal to 0.

The top subplot of Figure 6 shows Adam-ARS learns the optimal policy faster than ARS and shows much less variance in the associated rewards.

C. Oscillation Attacks

An attack on 30% of DER VV/VW controllers that creates voltage oscillations is depicted in Fig. 7. This baseline case demonstrates the effect of the attack in the system without utilizing the policy trained by Adam-ARS to control the remaining DER smart inverters in the system. Fig. 8 shows the effect of a *linear* policy trained by Adam-ARS. Clearly, the trained policy is effective in adjusting DER smart inverter VV/VW controllers to minimize voltage oscillations in the network shortly after they first manifest. Hyperparameters for this experiment are shown in Table I.

D. Imbalance Attacks

While a linear policy is sufficient to mitigate oscillation attacks, mitigation of imbalance attacks requires a more

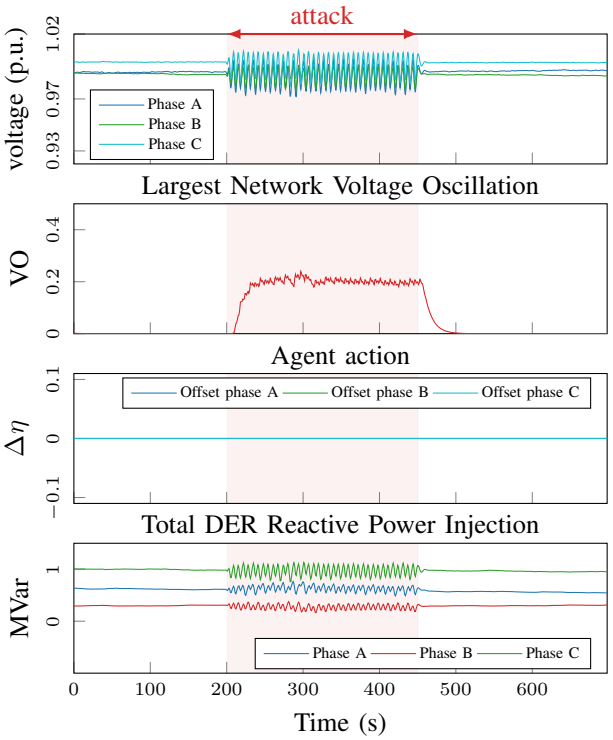

Fig. 7: 30% DER oscillation attack with no defense

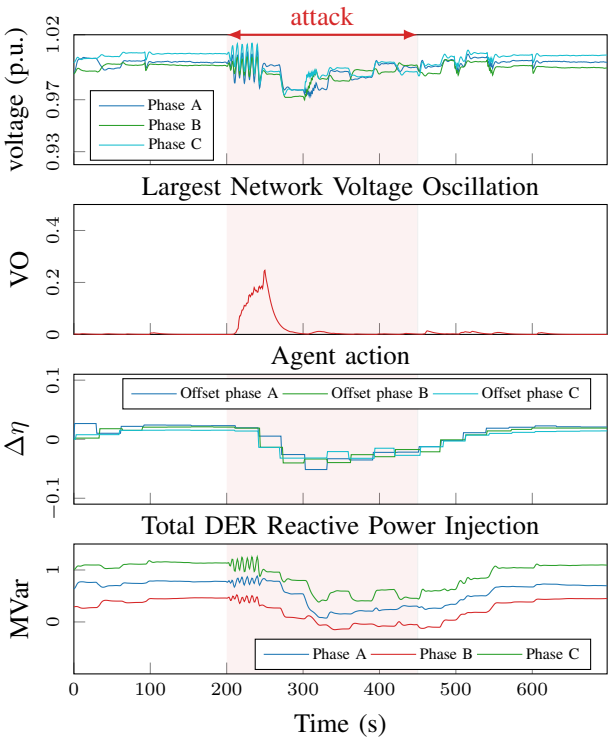

Fig. 8: 30% DER oscillation attack with ADAM-based ARS

complex policy structure (i.e., a neural network). An attack on 30% of DER VV/VW controllers aimed at creating large voltage imbalances is depicted in Fig. 9. This baseline case demonstrates the effect of the attack in the system without utilizing the policy trained by Adam-ARS to control the remaining DER smart inverters in the system. Fig. 10 shows the effect of a neural network-based policy trained by Adam-ARS. As is shown in the figure, the policy significantly reduces the largest voltage imbalance from 3% to less than 1%. Hyperparameters for this experiment are shown in Table II.

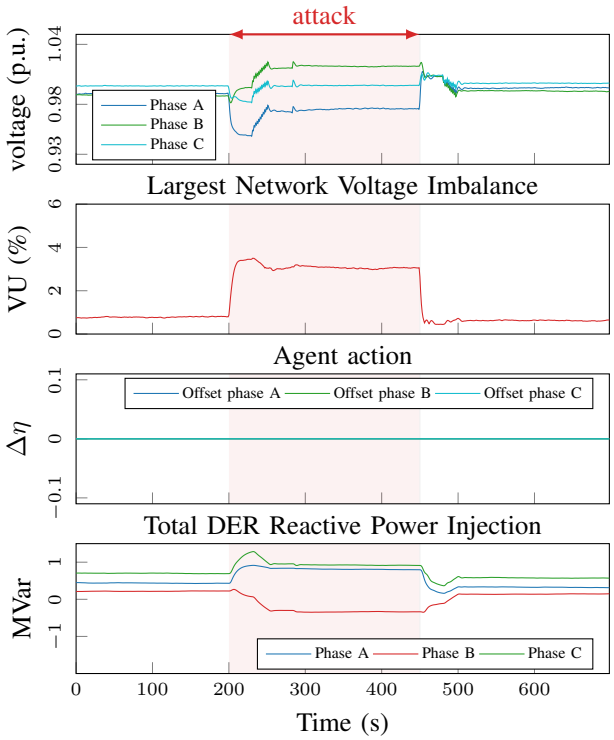

Fig. 9: 30% DER imbalance attack with no defense

V. Conclusions

This paper explored the use of an Adam-based Augmented Random Search (ARS) algorithm to directly search for optimal policies that manage DER smart inverter Volt-VAR/Volt-Watt control functions to mitigate the effects of cyber attacks. In the event that a malicious entity gains the ability to manipulate the VV/VW settings in a portion of DER in a given system, the trained optimal policy was shown to be effective in reconfiguring the remaining *non-compromised* DER VV/VW controllers in the system to mitigate large oscillations and large imbalances in system voltages. Compared to previous attempts, we demonstrated that the Adam-ARS approach can learn optimal policies significantly faster than Proximal Policy Optimization (PPO), faster than ARS, and with less variance in the reward compared to ARS. Additionally, the Adam-ARS algorithm is able to learn a *linear* policy for defense against voltage oscillation attacks. The results herein indicate that training optimal control policies for DER cybersecurity can be performed with fewer computational resources than previously believed, thereby allowing grid-managing entities with less expertise and financial resources the ability to use this technology for their systems.

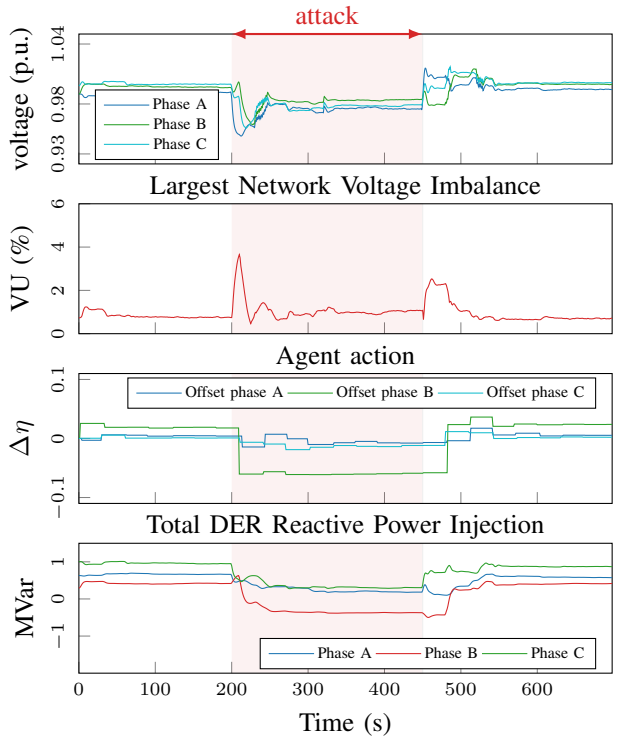

Fig. 10: 30% DER imbalance attack with ADAM-based ARS

While in this work we have demonstrated the effectiveness in adapting the learning rate during the training process, future work will explore adapting the variance of the exploration noise used for finite-difference approximation of the reward gradient. We believe that this feature may further reduce training time and possibly yield superior rewards compared to the proposed algorithm.

REFERENCES

[1] "Proposed Clean Energy Standard Could End Power Plant Greenhouse Gas Emissions By 2035," https://www.npr.org/2021/08/11/1026831067/proposed-clean-energy-standard-could-end-power-plant-greenhouse-gas-emissions-by, [Online; published Aug. 11, 2021].

[2] "DOE Announces Goal to Cut Solar Costs by More than Half by 2030," https://www.energy.gov/articles/doe-announces-goal-cut-solar-costs-more-half-2030, [Online; published Mar. 25, 2021].

[3] IEEE Standards Coordinating Committee 21, "IEEE Standard for Interconnection and Interoperability of Distributed Energy Resources with Associated Electric Power Systems Interfaces," *IEEE Std 1547-2018 (Revision of IEEE Std 1547-2003)*, pp. 1–138, April 2018.

[4] S. Sahoo, T. Dragičević, and F. Blaabjerg, "Cyber Security in Control of Grid-Tied Power Electronic Converters–Challenges and Vulnerabilities," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 2019.

[5] "800,000 Microinverters Remotely Retrofitted on Oahu in One Day," https://spectrum.ieee.org/energywise/green-tech/solar/in-one-day-800000-microinverters-remotely-retrofitted-on-oahu, [Online; accessed June-2019].

[6] C. Roberts, S.-T. Ngo, A. Milesi, S. Peisert, D. Arnold, S. Saha, A. Scaglione, N. Johnson, A. Kocheturov, and D. Fradkin, "Deep Reinforcement Learning for DER Cyber-Attack Mitigation," in *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, 2020, pp. 1–7.

[7] C. Roberts, S.-T. Ngo, A. Milesi, A. Scaglione, S. Peisert, and D. Arnold, "Deep Reinforcement Learning for Mitigating Cyber-Physical DER Voltage Unbalance Attacks," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 2861–2867.

[8] J. C. Spall, *Introduction to Stochastic Search and Optimization*. John Wiley & Sons, Ltd, 2003.

[9] M. Krstic, *Real-Time Optimization by Extremum-Seeking Control*. John Wiley & Sons, Ltd, 2003.

[10] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, "Evolution Strategies as a Scalable Alternative to Reinforcement Learning," https://arxiv.org/abs/1703.03864, 2017.

[11] H. Mania, A. Guy, and B. Recht, "Simple random search provides a competitive approach to reinforcement learning," https://arxiv.org/abs/1803.07055, 2018.

[12] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. 61, pp. 2121–2159, 2011.

[13] B. Seal, "Common Functions for Smart Inverters, 4th Ed." Electric Power Research Institute, Tech. Rep. 3002008217, 2017.

[14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," https://arxiv.org/abs/1412.6980, 2017.

[15] G. Hinton, N. Srivastava, and K. Swersky, "Lecture notes on Neural Networks for Machine Learning - Lecture 6a."

[16] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, Y. W. Teh and M. Titterington, Eds., vol. 9. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 249–256.

APPENDIX

Hyperparameter	**ARS**	**ADAM-based ARS**	**PPO**
α	1×10^{-2}	5×10^{-2}	1×10^{-3}
μ	3×10^{-2}	3×10^{-2}	-
β_0	0.9	0.9	-
β_1	0.999	0.999	-
λ_{ADAM}	1×10^{-8}	1×10^{-8}	-
γ	–	–	0.5
λ_{PPO}	–	–	0.95
ϵ	–	–	0.1
episodes	16	16	8
activation function	tanh	tanh	tanh
hidden layers	dense(16,16)	dense(16,16)	dense(16,16)
σ_y	300	300	300
σ_u	300	300	300
σ_a	0.5	0.5	0.5
σ_0	1	1	1
σ_p	1	1	1

TABLE I: Hyperparameters of the network, training and reward for unbalance attack

Hyperparameter	**ARS**	**ADAM-based ARS**	**PPO**
α	3×10^{-3}	3×10^{-3}	1×10^{-3}
μ	1×10^{-2}	1×10^{-2}	-
β_0	0.9	0.9	-
β_1	0.999	0.999	-
λ_{ADAM}	1×10^{-8}	1×10^{-8}	-
γ	–	–	0.5
λ_{PPO}	–	–	0.95
ϵ	–	–	0.1
episodes	8	8	8
activation function	–	–	tanh
hidden layers	linear	linear	dense (16, 16)
σ_y	300	300	300
σ_u	300	300	300
σ_a	0.5	0.5	0.5
σ_0	1	1	1
σ_p	1	1	1

TABLE II: Hyperparameters of the network, training and reward for oscillation attack