International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Evasion-Aware Botnet Attack Detection using Deep Reinforcement Adversarial Learning

Ketan Anand[1], Raja Rao Budaraju[2], Sumit Kumar[3], B. Madhav Rao[4], Basant Sah[5]

Submitted: 15/09/2023 **Revised:** 27/10/2023 **Accepted:** 08/11/2023

Abstract: Adversarial evasions represent contemporary challenges for applications relying on Machine Learning (ML). The susceptibility of traditional ML inference systems introduces vulnerabilities that make botnet detectors susceptible to attacks through adversarial examples. Complex AI models and sophisticated attack techniques can be used to generate the evasions. One of the potential sources of evasion assaults is generative AI models. The lack of data, which causes ML classifiers to train with a bias toward samples from the majority of classes, is a serious concern as well. This paper proposed a novel "Deep Reinforcement Learning based Evasion Generative Adversarial Network" (DRLEVGAN) to protect evasion attacks and retain semantics of the attack sample. The proposed model also tackles the issues of data imbalance, evasion awareness, and maintaining functionality in the context of synthetic botnet traffic generation. This model does not need adversarial training for the machine learning classifiers since it can act as an adversarial-aware botnet detection model. DRLEVGAN demonstrates superior performance when compared to similar models such as "Auxiliary Classifier GAN (ACGAN) and Evasion Generative Adversarial Network (EVGAN)".

Keywords: Deep reinforcement learning, Botnet, EVGAN, ACGAN, DRLEVGAN

1. Introduction

The growth of adversarial risks in today's more connected and data-driven world has ushered in a new era of difficulties for systems that rely on machine learning (ML). Particularly in the area of botnet detection, the dependability of conventional ML inference systems has been questioned due to their susceptibility to adversarial attacks. Adversarial evasions, a modern and formidable adversary that poses a danger to the security of systems across numerous sectors, have arisen and are defined by their capacity to exploit flaws in ML models [1].

This paper delves into the realm of Adversarial Evasion-Aware Botnet Detection using Deep Reinforcement Learning, addressing the critical need to fortify ML-based botnet detectors against increasingly sophisticated adversarial threats. It is imperative to understand that botnets, which are networks of compromised devices orchestrated to perform malicious activities, have become a pervasive threat in cyberspace [2]. Consequently, the development of effective and resilient botnet detection systems is of paramount importance in safeguarding digital ecosystems.

Adversarial evasions are manipulative techniques devised to exploit vulnerabilities in ML models, particularly those used in botnet detection [3]. Traditional ML systems, although highly effective in many scenarios, have proven to be susceptible to these attacks, rendering botnet detectors prone to manipulation and evasion [19]. Adversaries armed with complex AI models and sophisticated techniques can generate evasive traffic that can circumvent conventional detection mechanisms.

One notable source of these evasion assaults is generative AI models, which are proficient at crafting traffic that mimics legitimate communication while concealing malicious intent [4]. This characteristic makes them particularly challenging adversaries for botnet detectors. Moreover, the scarcity of relevant training data exacerbates the problem. The lack of diverse and representative data leads ML classifiers to train with a bias toward the majority class, causing them to struggle when identifying rare or minority-class events, such as specific types of botnet activity.

In response to these pressing challenges, this paper introduces a novel and innovative solution: the Deep Reinforcement Learning-based Evasion Generative Adversarial Network (DRLEVGAN). This model is specifically designed to fortify botnet detectors against evasion attacks while simultaneously preserving the semantics of attack samples. It is noteworthy that DRLEVGAN operates without necessitating adversarial training for the machine learning classifiers, marking a significant departure from conventional approaches. Instead, it serves as an adversarial-aware botnet detection model, detecting and thwarting evasion attempts in real time.

[1]*Assistant Professor, Dept of CSE(AI&ML), Sreenidhi Institute of Science and Technology, Yamnempet, Ghatkesar, Hyderabad*
Email: ketananand13@gmail.com
[2]*Senior Member of Technical staff, Oracle 3990, Scottfield street Dublin 94568, CA, USA*
Email: rajaraob@yahoo.com
[3]*Associate Professor & Head, Dept of Artificial Intelligence & Machine Learning, COER University, Roorkee.*
Email: mail2dr.sumit@gmail.com
[4]*Associate Professor, Department of CSE, SIR C R Reddy College of Engineering, Eluru, AP.*
Email: madhavraob@gmail.com
[5]*Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India.*
Email: basantbitmtech2008@gmail.com

The core innovation of DRLEVGAN lies in its foundation of Deep Reinforcement Learning, a subfield of ML renowned for its prowess in handling complex decision-making tasks. By integrating this reinforcement learning paradigm with the principles of Generative Adversarial Networks (GANs), DRLEVGAN possesses the capability not only to generate synthetic botnet traffic but also to adapt and evolve its evasion tactics in response to emerging threats. This adaptability is of paramount importance in the dynamic landscape of cybersecurity, where adversaries perpetually devise new and sophisticated methods to evade detection.

In addition to its evasion-aware capabilities, DRLEVGAN addresses the persistent issue of data imbalance that plagues many ML classifiers. By strategically sampling and generating synthetic minority-class samples, DRLEVGAN mitigates bias during training, thereby enhancing the model's proficiency in detecting rare or minority-class events, which are often indicative of botnet activity [5,6]. The pursuit of Adversarial Evasion-Aware Botnet Detection using Deep Reinforcement Learning is a testament to the ever-evolving landscape of cybersecurity. Adversarial threats continually push the boundaries of ML-based systems, necessitating innovative and resilient approaches like DRLEVGAN.

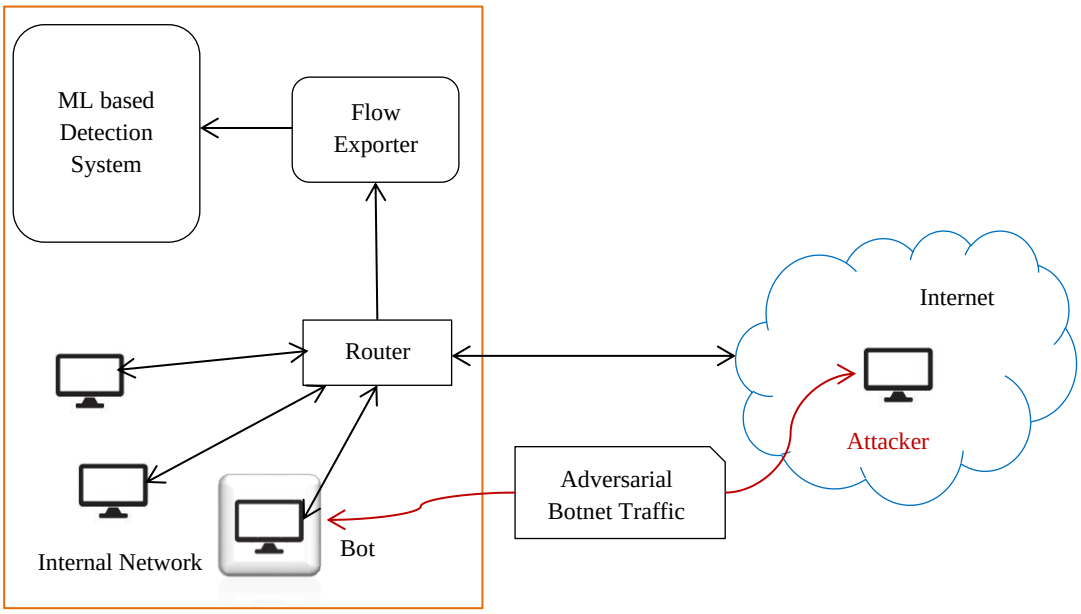

Fig 1. Threat Scenario Framework

Figure 1 demonstrates the framework of threat scenarios on the internet. Adversarial botnet traffic occurs between the internet attacker and the network bot. This research not only reinforces the critical importance of robust botnet detection but also highlights the potential of cutting-edge technologies, such as deep reinforcement learning and generative adversarial networks, in fortifying the security and reliability of ML-based systems. The adoption of our suggested methodology results in the creation of a system capable of both challenging cutting-edge botnet detectors and safeguarding them against both familiar and innovative evasion tactics. In contrast to previous efforts, our system ensures the security of these detectors even when confronted with unexpected evasion techniques, all while maintaining a high detection rate in the absence of adversarial attacks.

2. Literature Review

In [7], an innovative approach utilizing deep reinforcement learning (DRL) was introduced for the purpose of bypassing detection mechanisms designed to identify malicious PDF files. This novel framework takes inspiration from the concept of rewards, empowering a reinforcement learning agent to create adversarial samples by iteratively altering malicious PDF files. Ultimately, this iterative process is geared towards automating the evasion of detection within predefined constraints. What sets this approach apart from the conventional adversarial learning methods typically employed in similar studies is the proposal of a DRL-based defense strategy aimed at countering this attack.

[8] Introduced MalRNN, an innovative deep learning approach that offers the capability to automatically create evasive variants of malware without being constrained by any of the aforementioned limitations. This approach employs an adversarial example generation procedure, in which a language model is trained through a generative sequence-to-sequence recurrent neural network to enhance malware binaries. MalRNN excels at bypassing three recent deep learning-based malware detection systems and surpasses the current benchmark method in terms of performance.

An innovative approach is presented in this study [9], introduced a novel method for generating hierarchical adversarial attacks (HAA). This method is specifically designed to implement a “level-aware black-box adversarial attack strategy aimed at graph neural network (GNN)-based intrusion detection within resource-constrained IoT systems”. The strategy involves the creation of a shadow GNN model, incorporating an intelligent mechanism utilizing a saliency map technique to craft adversarial examples. These examples are generated with a focus on identifying and

modifying critical feature elements while minimizing perturbations.

To enhance the effectiveness of the attack, a hierarchical node selection algorithm is developed. This algorithm is grounded in a random walk with restart (RWR) approach and is responsible for identifying a set of nodes within the IoT network that are more susceptible to attacks due to their structural characteristics.

In [10], AppCon was introduced to counteract the impact of subtle alterations to malicious samples, serving as an innovative strategy for bolstering intrusion detectors against adversarial evasion attacks. This approach harnesses ensemble learning within real-world network environments, combining layers of specialized detectors for monitoring application behavior. Extensive experiments in diverse network scenarios, simulating botnet detection, involving detectors based on various machine and deep learning algorithms validate AppCon's effectiveness. It successfully thwarts over 75% of evasion attempts, without the drawbacks seen in existing countermeasures, such as performance degradation in non-adversarial contexts.

Numerous smart intrusion detection systems have been suggested in [11] for IoT settings, aiming to achieve both superior detection accuracy and meet high-level prerequisites. These objectives are intertwined with the optimization of computational resources, minimizing power consumption at the edge, and safeguarding the privacy of sensitive data. Their goal is not solely centered on presenting a comprehensive classification of design choices made by DRL experts in the realm of intrusion detection. We also aim to delve into the benefits and practical implementation of each configuration within real IoT environments.

[12] Introduced a Reinforcement Learning (RL) based solution designed to uncover malicious payloads capable of circumventing Web Application Firewalls (WAFs). We present an RL framework conforming to the OpenAI Gym toolset standards, creating an environment for training agents in WAF circumvention tasks. These payloads, adept at bypassing WAFs, reveal rule deficiencies that can subsequently inform rule-based WAF tuning. Furthermore, they contribute to enriching datasets for retraining machine learning-based WAFs. Leveraging Q-Learning, Advantage Actor-Critic (A2C), and Proximal Policy Optimization (PPO) algorithms in conjunction with deep neural networks, our solution effectively evades both signature-based and machine learning-based WAFs.

3. Methodology

The effectiveness of the DRL attacker has a positive impact on EVAGAN's performance. The generation of samples using GANs adheres to the probability distribution of the input data, with the GAN itself adjusting based on feedback from Ď. η, in its pursuit to explore unfamiliar sample spaces that can deceive Ď, occasionally produces samples that do not accurately reflect the genuine malicious patterns. To overcome this challenge, the utilization of DRL samples can be instrumental in enabling η to discern the boundaries of authentic samples. Consequently, a "DRL agent" [20] can be employed to explore samples within a well-defined observation space. Through training Ď on these newly developed samples, feedback is looped back into the GAN training process.

In order to align with the parameters established by the DRL agent, progressively modifies and fine-tunes its sample creation, ultimately improving the fidelity of the samples created. This procedure can expedite the convergence of η's training while simultaneously attaining a high level of accuracy in a reduced number of epochs. While it is true that enhancing semantic awareness incurs extra training expenses for the DRL component, this justification serves as the driving force behind the advancement represented by DRLEVGAN towards a more intelligent GAN design that prioritizes the preservation of functionality.

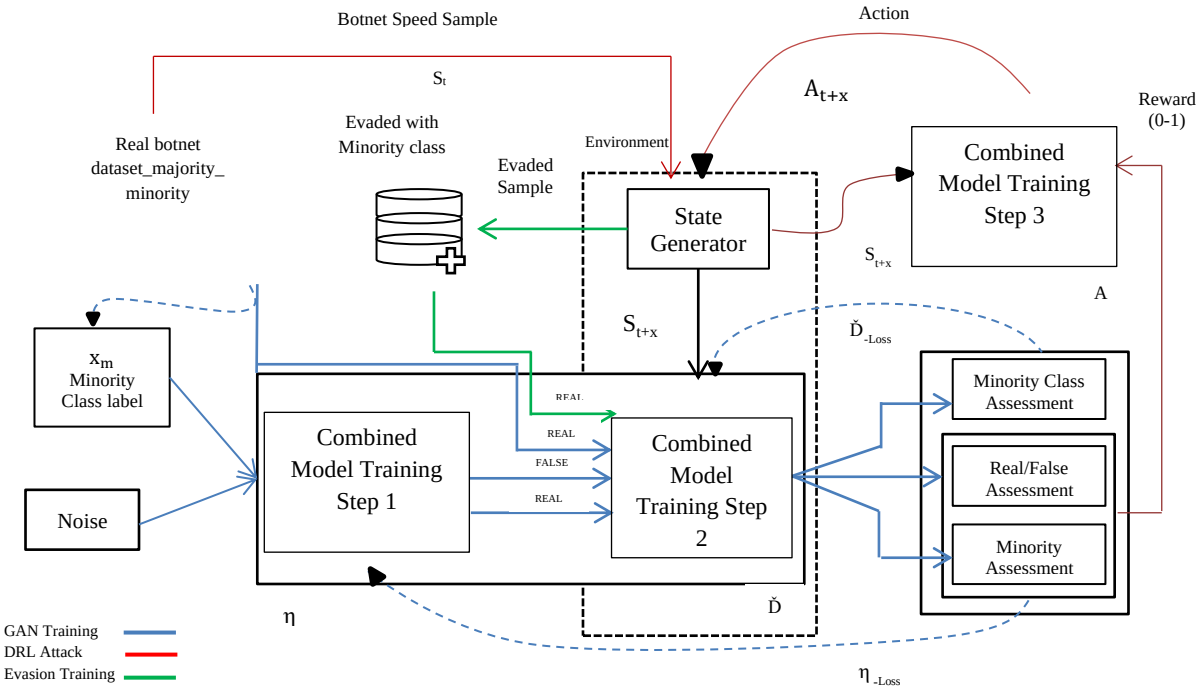

Fig 2. Architecture of DRLEVGAN

Figure 2 illustrates the intricate architecture of DRLEVGAN, which is comprised of two key components: EVGAN and a DRL model [7]. Similar to other DRL-based attackers designed for evading detection in "black-box" attack scenarios [8-10], DRLEVGAN's DRL agent targets Ď within EVGAN, serving as a "black-box classifier". Ď's output, specifically its estimation of the minority class, is harnessed as a reward signal to guide the agent in fine-tuning its weights and generating a novel action denoted as A_{t+x} based on a predefined policy ρ.

This freshly generated action is then introduced into the environment, where the state generator leverages another seed sample sourced from the authentic dataset to craft a new state. In light of receiving a optimistic reward, the evasion samples are systematically fed into Ď of EVGAN, facilitating its adversarial training. Through this iterative process, Ď becomes actively attuned to potential future evasion attempts, thereby enhancing its ability to provide more informative feedback to η, the core element responsible for constraining the evasion generation boundaries. This systematic approach results in the early convergence of η's training, contributing to the overall effectiveness of DRLEVGAN in evading detection while concurrently strengthening the model's proactive defense against future evasion attempts.

State space and action space are the two main parts of the environment in DRLEVGAN [11]. The raw bytes of the botnet flow sample are present within the environment. To better and more efficiently represent the state of these samples, it becomes essential to express them in the form of a feature vector. Given the extended length and intricate nature of each flow, we employ autoencoders (AE) to transform these flows into concise binary codes. An autoencoder is a specific type of feedforward neural network employed for unsupervised learning, primarily utilized for tasks like dimensionality reduction and feature extraction.

Its role is to capture essential information from the input data and encode it into a more compact representation, which is particularly useful in scenarios where simplifying the data is advantageous. In our case, this process aids in creating shorter binary codes that efficiently encapsulate the state of the botnet flow sample, making it more manageable and suitable for subsequent analysis and processing.

To prepare the flows for encoding, they are first trimmed to 1024 bytes, effectively rendering them as grayscale images sized 32x32. If a flow falls short of this length, '0x00' bytes are appended to the end. Typically, the initial bytes of a flow contain vital connection information and a limited amount of content exchange data, making them more reflective of the flow's characteristics. Subsequently, we employ a convolutional auto-encoder, utilizing CNNs both as the encoder and decoder, to convert this "image" into a concise 32-dimensional feature vector. This resultant vector offers a comprehensive representation of the botnet flow sample and serves as the state within the Markov Decision Process (MDP).

In order to maintain the integrity of functionality, any alteration to the feature value is constrained to $\boldsymbol{\delta}$, defined as the smallest value among the specific feature values within the dataset, as outlined in Equation 1.

$$\delta_x = \min_{F_x \exists \Delta l} x(l) \qquad (1)$$

Where, F_x represents the smallest value observed among all rows (denoted as 'l') corresponding to a specific feature 'F.' In this context, 'x' denotes the action index originating from the agent and signifies a particular feature number extracted from the action table.

The agent in RELEVAGAN is a deep neural network that takes the size of the observation space as its input and generates the number of actions as its output. In the context of DRLEVGAN, the observation space represents a comprehensive botnet sample represented as a feature vector. The primary role of the agent is to decide by selecting an action index 'x' from the available features.

DRLEVGAN Training Algorithm

Step 1: Loop over batches from 1 to the total number of sets

Step 2: Train Discriminator (Ď) using real data

Step 3: Train Discriminator (Ď) using generated data

Step 4: Utilize the DRL Agent (A) to generate evasions for the current batch size

Step 5: Train Discriminator (Ď) using the evasions generated by the DRL Agent (A)

Step 6: Train Generator (η)

Step 7: End of loop

DRLEVGAN closely resembles EVGAN, but it incorporates a series of additional steps involving a Deep Reinforcement Learning (DRL) agent. These extra stages, specifically Step 4 and Step 5, distinguish the training process of EVGAN from DRLEVGAN within the algorithm. The precise sequence of these steps is paramount in comprehending the underlying logic of DRLEVGAN's operation.

In the described algorithm, Step 4 and Step 5 play a pivotal role in shaping the divergence between EVGAN and DRLEVGAN, and this distinction persists for a predetermined number of batches during training. These steps establish a dynamic interplay between the generative model, the discriminator, and the DRL agent, infusing DRLEVGAN with unique learning dynamics.

It's noteworthy that the training of η, a crucial component of the system, occurs after the evasion training of Ď. This sequential arrangement is instrumental in fine-tuning the performance of Ď. As Ď's weights are continually adjusted based on the evasions developed by the DRL attacker in Step 5, it acquires a heightened awareness of evasion strategies. Consequently, when Ď provides feedback in the form of η_loss to η, this feedback is informed and strategic, substantially enhancing the training process compared to the conventional training of EVGAN.

To emphasize, the DRL attack is executed after each training batch, consistently infusing adversarial elements into the training process, and facilitating the evolution of DRLEVGAN's evasion capabilities over time. This dynamic interplay between DRL and generative adversarial techniques renders DRLEVGAN a formidable tool for evasion generation and defense.

In a conventional model for generating botnet evasions using a DRL black-box attack, the reward is typically determined by the

outcome of the botnet detector [12], which is typically a actual number within [0, 1] range. In the scenario presented here, when assessing a botnet sample, the expected reward value is "0", and ideally, for a standard traffic sample, the reward should be "1". Consequently, to classify a botnet sample as an effective evasion accomplished by a DRL assailant, a threshold is established where the reward must surpass 0.6. This threshold serves as a crucial benchmark for distinguishing between effective botnet sample evasions and other instances in the evaluation process.

4. Results & Discussion

The performance evaluation of DRLEVGAN aligns with that of EVGAN, especially when dealing with botnet datasets. Several key metrics were employed to gauge the effectiveness of the models, encompassing criteria such as the validity of generated samples (Gen_Val), the detection of false or generated botnet samples evasions (False_Bot_Ev), estimation accuracy of real "normal/majority" class (Real_Nor_Est), and the ability to detect evasions in real "botnet/minority" class samples (Real_Bot_Ev). These metrics are quantitatively represented through mathematical expressions denoted by Equations 2 to 5.

$$\text{Gen\_Val} = \frac{\text{Sum of } |\acute{\eta}(q, x_m)[0]|}{n} \quad (2)$$

In equation (2), $\acute{\eta}(q, x_m)[0]$ represents the prediction of whether the source is genuine or false. The closer this value approaches '1,' the stronger the inclination toward categorizing it as authentic. Where q is the noise sample, x_m is the minority class label.

$$\text{False}_{\text{Bot}_{\text{Ev}}} = \frac{\text{Sum of } |\acute{\eta}(q, x_m)[1]|}{n} \quad (3)$$

In equation (3), $\acute{\eta}(q, x_m)[1]$ signifies the likelihood estimation for developed minority/botnet class samples. Given that the label for the "minority/botnet" class is designated as '0,' an optimal scenario would involve the model producing a value proximate to '0.' This estimation is indicative of the level of evasion observed in the generated samples. Therefore, the closer this value approaches '0,' the lower the degree of evasion detected.

$$\text{Real\_Nor\_Est} = \frac{\text{Sum of } |R_{Normal.test}[2]|}{n} \quad (4)$$

In equation (4), $R_{Normal.test}[2]$ denotes the probability assessment associated with "majority/normal" class samples. Given that the majority/normal class is identified by the label '1,' the model is ideally anticipated to generate a value in proximity to '1' for this estimation. Where R represents the real data distribution.

$$\text{Real}_{\text{Bot}_{\text{Ev}}} = \frac{\text{Sum of } |R_{Botnet.test}[1]|}{n} \quad (5)$$

In equation (5), $R_{Botnet.test}[1]$ signifies the probability estimation about actual minority/botnet class samples.

4.1 Dataset Preparation

For experimentation, two datasets are used in this paper. CIC-IDS-2017 and CIC-IDS-2018 from the Canadian Institute of Cybersecurity (CIC). The CICIDS2017 research initiative made the deliberate choice to focus its attention on the Ares botnet. In pursuit of this objective, comprehensive network traffic data pertaining to the Ares botnet was meticulously collected within the confines of the CIC facility. This dataset is readily accessible for reference and analysis on the official CIC website.

To facilitate a more streamlined and specific analysis, a distinct subset of this dataset was created. This subset exclusively encompasses all normal network flows, and it is thoughtfully combined with the selected botnets for a more refined examination. This approach enables researchers and analysts to delve deeper into the behavior and characteristics of Ares and its interaction with regular network traffic.

In the pursuit of generating additional subsets from an inherently unbalanced dataset, CIC-IDS2018 dataset is considered. This particular dataset provided with samples representing two significant botnets, Ares and Zeus. This approach involved the careful curation of a new subset, comprised of not only the regular network traffic but also a carefully selected more than 2000 traffic flows associated with the botnets.

The deliberate inclusion of these botnet traffic flows further amplified the imbalance within this dataset. This new, unbalanced dataset serves as a valuable resource for in-depth analysis, enabling researchers to explore and understand the intricacies of Ares and Zeus botnet behavior within a broader network context.

The generator η of EVGAN only takes noise q and the single class labels $x_m = 1$ as the only binary classification is considered. The labels are embedded in the input layer of η. The objective function of η has two parts as shown in Equations (6) and (7).

$$X^{\eta}(\eta) = \sum[log\,(\check{D}(\eta(q)))] \quad (6)$$

$$Y^{\eta}(\eta) = \sum\left[logR\left(X = x_m|\sigma_{mfalse}\right)\right] \quad (7)$$

Equation (6) is the objective function of η. The goal is to minimize the log-likelihood of the false samples being classified as false by Ď.

In equation (7), $Y^{\eta}(\eta)$ is the objective function of η for improving the log-likelihood of minority class samples coming from η into Ď, where R is the output probability from Ď. Since η only needs to generate x_m samples so it should only receive the loss of Ď on the estimation of minority class and the sources, i.e. the samples being real or false. The objective function of η is to maximize the Ď loss on the false source.

At the same time, it will assist in minimizing Ď loss on x_m samples. Equation (8) shows the objective function of η.

$$L^{\eta}(\eta) = Y^{\eta}(\eta) - X^{\eta}(\eta) \quad (8)$$

The objective function of Ď has three parts as given by Equations (9), (10), and (11). For the minority class, "m" was used in the following equations,

$$L_M = \sum[\text{logR}(X = X_m|\sigma_{mreal})] \quad (9)$$

$$L_{\sigma_M} = \sum[log\,R(\varepsilon = real|\sigma_{mreal})] + \sum[log\,R(\varepsilon = real|\sigma_{mfalse})] \quad (10)$$

$$L_m = \sum[log\,R(X = X_m|\sigma_{mreal})] + \sum[log\,R\left(X = X_m|\sigma_{mrfalse}\right)] \quad (11)$$

At its inception, the primary objective of 'Ď' is to accurately gauge the distribution of the majority class by relying solely on

the real samples. This is particularly crucial because 'η' is not responsible for generating any of the majority class samples. In this context, Equation (9) plays a pivotal role, portraying the source log-likelihood associated with both genuine and fabricated minority class samples. Equation (10) then succinctly consolidates the log-likelihoods for both real and fabricated instances from 'Ď,' specifically pertaining to the minority class.

As a result, the overarching aim of 'Ď' is to optimize its objective function, and this can be aptly depicted as the collective maximization of three distinct log-likelihoods. These log-likelihoods are detailed in Equation (11) and encompass the genuine and counterfeit minority class samples' log-likelihoods and the real instances from the majority class. This comprehensive approach ensures that 'Ď' is well-equipped to effectively estimate the majority class distribution while navigating the intricacies of class-specific log-likelihoods.

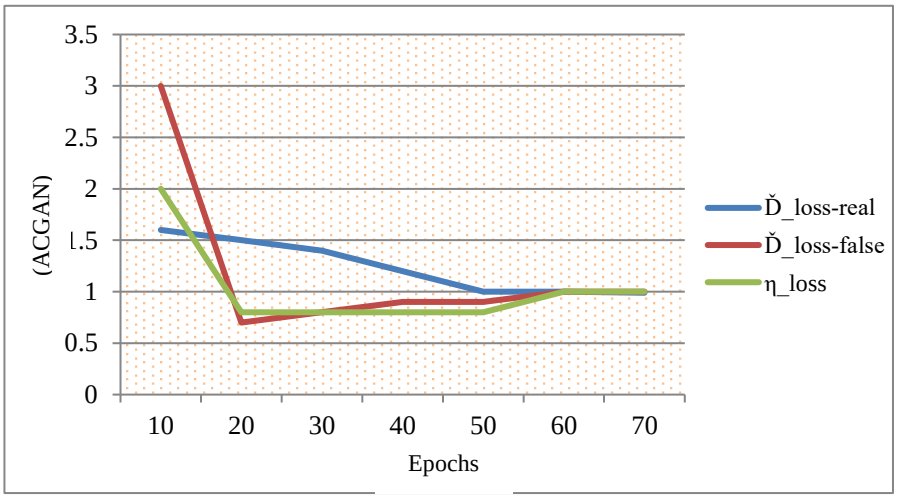

Fig 3. Training loss Scenario (ACGAN)

To provide a visual representation of the evaluation process, Figures 3, 4, and 5 depict the loss profiles of Ď for both real and false minority classes, as well as "majority/normal" classes, and the loss profiles of η for models including ACGAN, EVGAN, and DRLEVGAN. These figures offer an insightful perspective into the performance characteristics of the models under consideration, aiding in the comprehensive analysis of their capabilities and effectiveness in addressing botnet detection and evasion challenges. Figure 3 demonstrates Ď and η losses for ACGAN.

The EVGAN serves as the fundamental model in this context, offering the distinct advantage of not necessitating the adversarial training of dedicated machine learning classifiers. This is primarily due to the inherent capability of EVGAN, where the 'Ď' component inherently functions as an evasion-aware botnet detector. Consequently, the DRLEVGAN, as an extension of EVGAN, likewise avoids the need for adversarial training. Nevertheless, the evasion tactics generated by the DRL attacker must be integrated into the 'Ď' component. This step essentially replicates the back-propagation process commonly seen in generative adversarial network training, thus ensuring that the system remains robust and well-prepared to identify and counteract evasion techniques effectively.

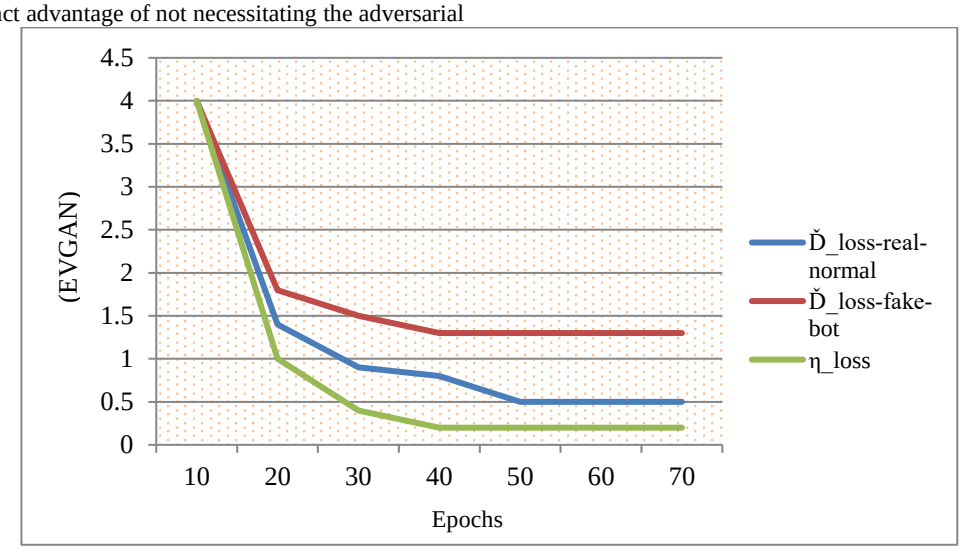

Fig 4. Training loss Scenario (EVGAN)

Figure 4 demonstrates Ď and η losses for EVGAN. Figure 5 demonstrates Ď and η losses for DRLEVGAN. It appears that the loss convergence pattern is consistent across all GANs. Notably, DRLEVGAN tends to reach its lowest point earlier than both EVGAN and ACGAN. It's worth mentioning that in the case of DRLEVGAN, the Ď_Loss_False values are relatively higher

compared to the other GANs. This discrepancy might be attributed to the challenge faced by the discriminator (Ď) in DRLEVGAN, which struggles to differentiate between evasions generated by the DRL agent and those from the generator (η). Since the evasion samples produced by the DRL agent are labeled as "REAL", when similar samples reach from η, the misclassification leads to increased loss.

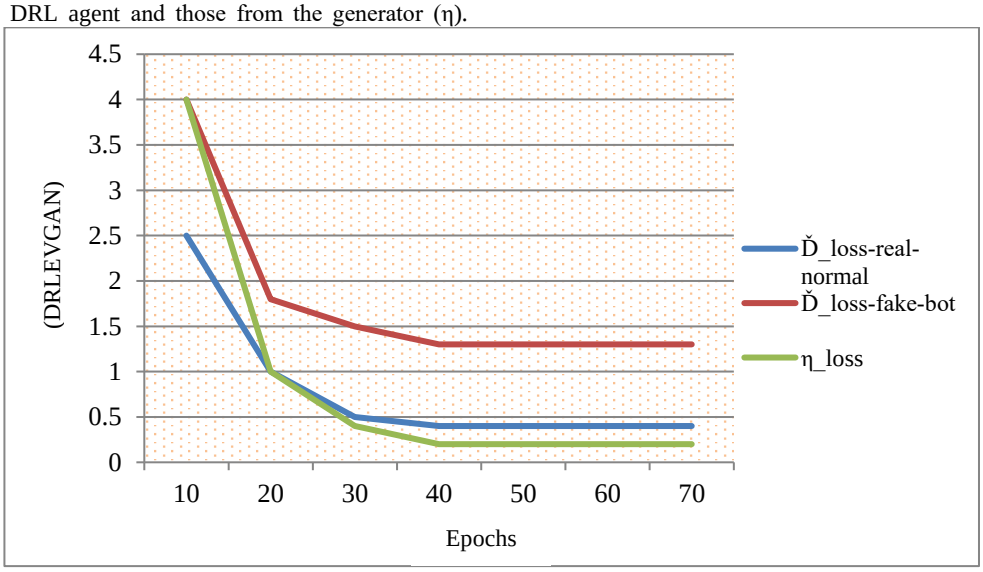

Fig 5. Training loss Scenario (DRLEVGAN)

However, it's important to note that the total detection performance enhances because η in RELEVAGAN tends to generate samples that adhere more closely to semantic constraints. While this phenomenon could be associated with mode collapse, it's essential to emphasize that the detection performance of RELEVAGAN surpasses that of EVAGAN, mitigating concerns related to mode collapse.

The early convergence and distinct loss patterns in RELEVAGAN, despite higher D Loss False values, indicate an effective approach that enhances detection performance by constraining sample generation within semantic boundaries.

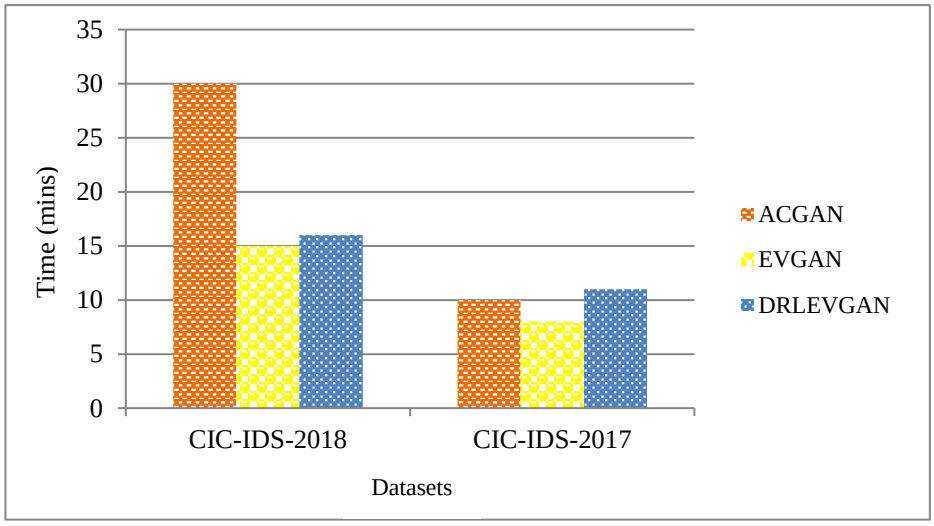

Fig 6. Time Complexity

Figure 6 displays the training time complexity comparison among ACGAN, EVGAN, and DRLEVGAN for the two datasets. The training time for RELEVAGAN over 100 epochs is slightly extended compared to EVGAN due to the additional computational overhead introduced by the DRL component.

Nonetheless, it's noteworthy that achieving 100% performance is accomplished in significantly less time than required by EVAGAN, although this timeframe appears to be contingent on the specific dataset in use.

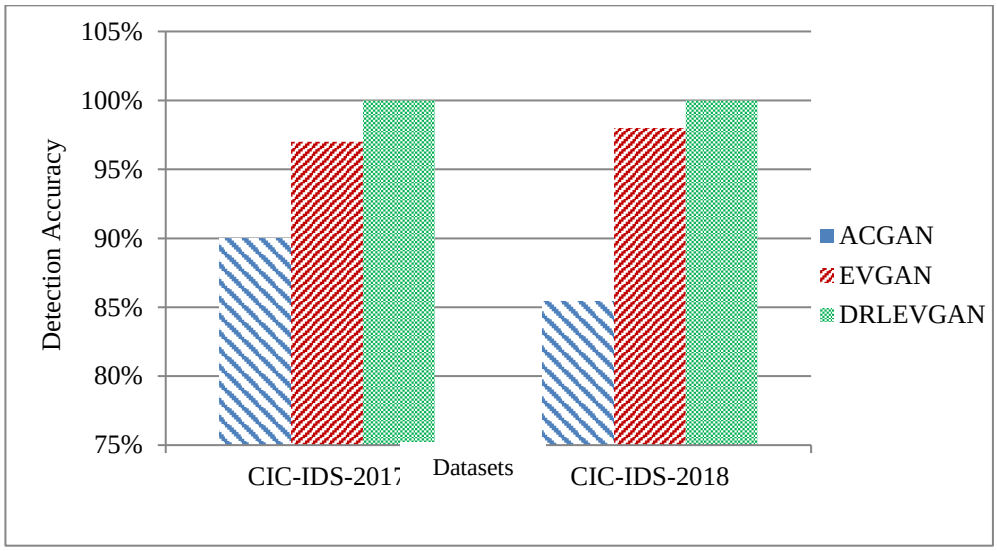

Fig 7. Performance Comparison

DRLEVGAN is an enhanced version of EVGAN, so it is best to compare its performance with EVGAN and ACGAN. For both datasets, DRLEVGAN gives detection accuracy of 100% as compared to ACGAN and EVGAN. DRLEVGAN serves as both an oversampling tool for adversarial evasion models and an evasion-aware finding model tailored for botnets in scenarios characterized by limited data availability.

A cutting-edge botnet finding model can leverage deep reinforcement learning to create adversarial evasion samples, thereby preserving the fundamental effectiveness of botnet and malware samples. When integrated with EVGAN's discriminator model, deep reinforcement learning aids the EVGAN generator in constraining the exploration space within semantic boundaries. The findings indicate that, in scenarios with limited data availability, DRLEVGAN, empowered by deep reinforcement learning, offers the advantage of achieving early convergence in botnet detection, outperforming EVGAN in this regard.

5. Conclusion

In the face of contemporary challenges posed by adversarial evasions in Machine Learning (ML) applications, this research offers a significant contribution. Traditional ML inference systems exhibit susceptibility, rendering botnet detectors vulnerable to adversarial attacks. These attacks, driven by intricate AI models and sophisticated techniques, underscore the pressing need for robust defense mechanisms. Among potential sources of evasion threats, generative AI models are noteworthy. This study introduces the innovative “Deep Reinforcement Learning based Evasion Generative Adversarial Network” (DRLEVGAN), designed to counter evasion attacks while preserving the semantic integrity of attack samples. Beyond this, DRLEVGAN addresses critical concerns related to data imbalance, evasion awareness, and the protection of functionality within synthetic botnet traffic generation.

Notably, DRLEVGAN distinguishes itself by functioning as an adversarial-aware botnet finding model without the necessity for adversarial training of machine learning classifiers. Our experimental results affirm the model's superior performance in comparison to similar models, including “Auxiliary Classifier GAN (ACGAN)” and “Evasion Generative Adversarial Network (EVGAN)”. In summary, this research offers a timely and effective solution to the contemporary challenge of securing ML systems against adversarial evasions, especially in the realm of botnet detection. DRLEVGAN's exceptional performance and evasion awareness capabilities promise to fortify the defense against adversarial attacks, marking a significant step forward in safeguarding ML applications.

References

[1] Kazmi, S., Aafaq, N., Khan, M. A., Khalil, M., & Saleem, A. (2023). From Pixel to Peril: Investigating Adversarial Attacks on Aerial Imagery through Comprehensive Review and Prospective Trajectories. IEEE Access.

[2] Djenna, A., Barka, E., Benchikh, A., & Khadir, K. (2023). Unmasking Cybercrime with Artificial-Intelligence-Driven Cybersecurity Analytics. Sensors, 23(14), 6302.

[3] Debicha, I., Cochez, B., Kenaza, T., Debatty, T., Dricot, J. M., & Mees, W. (2023). Adv-Bot: Realistic adversarial botnet attacks against network intrusion detection systems. Computers & Security, 129, 103176.

[4] Neupane, S., Fernandez, I. A., Mittal, S., & Rahimi, S. (2023). Impacts and Risk of Generative AI Technology on Cyber Defense. arXiv preprint arXiv:2306.13033.

[5] Apruzzese, G., Andreolini, M., Marchetti, M., Venturi, A., & Colajanni, M. (2020). Deep Reinforcement Adversarial Learning Against Botnet Evasion Attacks. IEEE Transactions on Network and Service Management, 17(4), 1975-1987.

[6] Mari, A. G., Zinca, D., & Dobrota, V. (2023). Development of a Machine-Learning Intrusion Detection System and Testing of Its Performance Using a Generative Adversarial Network. Sensors, 23(3), 1315.

[7] Jiang, T., Liu, Y., Wu, X., Xu, M., & Cui, X. (2023). Application of deep reinforcement learning in attacking and protecting structural features-based malicious PDF detector. Future Generation Computer Systems, 141, 325-338.

[8] Ebrahimi, M., Zhang, N., Hu, J., Raza, M. T., & Chen, H. (2020). Binary black-box evasion attacks against deep

learning-based static malware detectors with adversarial byte-level language model. arXiv preprint arXiv:2012.07994.

[9] Zhou, X., Liang, W., Li, W., Yan, K., Shimizu, S., Kevin, I., & Wang, K. (2021). Hierarchical adversarial attacks against graph-neural-network-based IoT network intrusion detection system. IEEE Internet of Things Journal, 9(12), 9310-9319.

[10] Apruzzese, G., Andreolini, M., Marchetti, M., Colacino, V. G., & Russo, G. (2020). AppCon: Mitigating evasion attacks to ML cyber detectors. Symmetry, 12(4), 653.

[11] Rizzardi, A., Sicari, S., & Porisini, A. C. (2023). Deep Reinforcement Learning for intrusion detection in Internet of Things: Best practices, lessons learnt, and open challenges. Computer Networks, 236, 110016.

[12] Hemmati, M., & Hadavi, M. A. (2022). Bypassing Web Application Firewalls Using Deep Reinforcement Learning. ISeCure, 14(2).

[13] Randhawa, R. H., Aslam, N., Alauthman, M., & Rafiq, H. (2022). Evasion generative adversarial network for low data regimes. IEEE Transactions on Artificial Intelligence.

[14] Giovanni Apruzzese, Mauro Andreolini, Mirco Marchetti, Andrea Venturi, and Michele Colajanni. Deep reinforcement adversarial learning against botnet evasion attacks. IEEE Transactions on Network and Service Management, 17(4):1975–1987, 2020.

[15] Mao, Z., Fang, Z., Li, M., & Fan, Y. (2022). EvadeRL: Evading PDF malware classifiers with deep reinforcement learning. Security and Communication Networks, 2022.

[16] Di Wu, Binxing Fang, Junnan Wang, Qixu Liu, and Xiang Cui. Evading machine learning botnet detection models via deep reinforcement learning. In ICC 2019-2019 IEEE International Conference on Communications (ICC), pages 1–6. IEEE, 2019.

[17] Couto, G. C. K., & Antonelo, E. A. (2023). Hierarchical Generative Adversarial Imitation Learning with Mid-level Input Generation for Autonomous Driving on Urban Environments. arXiv preprint arXiv:2302.04823.

[18] Anderson, HS., Kharkar, A., Filar, B., Evans, D., & Roth, P. (2018). Learning to Evade Static PE Machine Learning Malware Models via Reinforcement Learning. arXiv:1801.08917 [cs.CR]

[19] Alauthman, Aslam, Al-Kasassbeh, Khan, Al-Qerem and Choo K K R 2020, "An efficient reinforcement learning-based botnet detection approach. J. Netw. Comput. Appl. 150, Article no 102479.

[20] Zhang, Q., Cho, J. H., Moore, T. J., Kim, D. D., Lim, H., & Nelson, F. (2023, May). EVADE: Efficient Moving Target Defense for Autonomous Network Topology Shuffling Using Deep Reinforcement Learning. In International Conference on Applied Cryptography and Network Security (pp. 555-582). Cham: Springer Nature Switzerland.