PDF Download
3664476.3669928.pdf
11 March 2026
Total Citations: 4
Total Downloads: 1025

Published: 30 July 2024

Citation in BibTeX format

ARES 2024: The 19th International Conference on Availability, Reliability and Security
July 30 - August 2, 2024
Vienna, Austria

RESEARCH-ARTICLE

ARGAN-IDS: Adversarial Resistant Intrusion Detection Systems using Generative Adversarial Networks

JOÃO COSTA

FILIPE APOLINÁRIO

CARLOS RIBEIRO, Higher Technical Institute, Lisbon, Lisbon, Portugal

Open Access Support provided by:

Higher Technical Institute

ARES '24: Proceedings of the 19th International Conference on Availability, Reliability and Security (July 2024)
https://doi.org/10.1145/3664476.3669928
ISBN: 9798400717185

ARGAN-IDS: Adversarial Resistant Intrusion Detection Systems using Generative Adversarial Networks

João Costa
joao.f.pereira.costa@tecnico.ulisboa.pt
INOV-INESC INOVAÇÃO
R. Alves Redol 9, 1000-029, Lisbon
Portugal

Filipe Apolinário
filipe.apolinario@tecnico.ulisboa.pt
INOV-INESC INOVAÇÃO
R. Alves Redol 9, 1000-029, Lisbon
Portugal

Carlos Ribeiro
carlos.ribeiro@tecnico.ulisboa.pt
INESC-ID, Instituto Superior Técnico,
Universidade de Lisboa
R. Alves Redol 9, 1000-029, Lisbon
Portugal

ABSTRACT

Neural Networks (NNs) are not secure enough to be deployed on security-critical tasks such as Network Intrusion Detection Systems (NIDS). NNs are vulnerable to Adversarial Attacks (AAs), which affect their accuracy in identifying malicious activity, by introducing perturbations on network traffic. This work proposes "Adversarial Resistant Intrusion Detection Systems using GANs" (ARGAN-IDS) a method to address these vulnerabilities. ARGAN-IDS is implemented as a Generative Adversarial Network (GAN) trained on network traffic to protect NIDS. ARGAN-IDS, greatly mitigates the impact of AAs, achieving comparable results to a non-perturbed execution. We show GANs have limitations in differentiating between malicious traffic and traffic altered by AAs. And we address this in ARGAN-IDS by training the GAN on network traffic containing malicious packets. This enhancement significantly improved the GAN's performance, enabling it to identify even highly perturbed adversarial attacks effectively. ARGAN-IDS acts as a neutralizer of perturbations introduced by AAs and mitigates the NIDS vulnerabilities. We have integrated ARGAN-IDS with a state-of-the-art anomaly-based detector, Kitsune. We achieve a reduction of 99.27% of false positives and an improvement of 99.29% of the true negatives, leading to an improvement of roughly 36.75% in overall system accuracy while under AAs.

KEYWORDS

Deep Neural Network (DNN), Adversarial Attacks (AA), Generative Adversarial Network (GAN), Network Intrusion Detection Systems (NIDS)

ACM Reference Format:
João Costa, Filipe Apolinário, and Carlos Ribeiro. 2024. ARGAN-IDS: Adversarial Resistant Intrusion Detection Systems using Generative Adversarial Networks. In *The 19th International Conference on Availability, Reliability and Security (ARES 2024), July 30–August 02, 2024, Vienna, Austria.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3664476.3669928

1 INTRODUCTION

Adversarial attacks on NNs have emerged as a major concern in the field of machine learning[5, 9, 10, 15, 19, 23, 24], with the potential to compromise the accuracy and integrity of these systems in a variety of applications, including security-critical tasks such as intrusion detection[1–3, 7, 8, 17, 20, 25]. In response, researchers have developed a range of techniques for improving the resilience of NNs against these attacks[6, 11, 12, 16, 18, 21, 22].

Despite this, research into AAs for NIDS is currently very scarce. While a majority of the effective defensive measures have been conceptualized and developed primarily for the image classification domain, the implications of AAs on A-NIDS cannot be overlooked. ARGAN-IDS focuses on the protection of anomaly-based NIDS (A-NIDS). Its operation is based on learning the representation of normal traffic, and any traffic that does not match this representation is considered anomalous and eventually triggers an alarm. In these systems, an attack could mean an increase in false positives, causing unwanted alarms, or an increase in false negatives. Both these factors result in a reduction of their reliability.

We add to the work previously done in the image classification domain and verify its applicability and intricacies when applied to an anomaly-based autoencoder, focusing on a reconstruction-based approach introduced by DefenseGAN[18]. Our ARGAN-IDS system is used as a *neutralizer* network which approximates an adversarial example by generating a new, similar but non-adversarial example. This is done using a GAN that is trained on non-adversarial data and generates new examples that are similar to the original ones. This generated example is given as input to the NIDS instead. By adding this defense layer to the state-of-the-art NIDS, Kitsune[13], we compare the added value and eventual tradeoffs this approach might introduce in the original system. We show that our ARGAN-IDS system is very capable of reducing the adversarial perturbations introduced on NIDS. Throughout our testing, we verify that our approach can reduce up to 99.27% of the introduced false positives and achieve an improvement of 99.29% of the true negatives, leading to a gain of roughly 36.75% in overall system accuracy while under adversarial attacks. We observe these results can be achieved at a cost of only a marginal decrease in accuracy of around 1% in a normal execution. This can be taken as a small tradeoff for the improved robustness of the system. Furthermore, we show that ARGAN-IDS can distinguish between malicious and adversarial traffic, making it a viable approach to detecting these attacks.

We contribute to the ongoing efforts to prevent already existing and new attacks from compromising the reliability of these systems. The main contributions of our work are:

ARES 2024, July 30–August 02, 2024, Vienna, Austria

ACM ISBN 979-8-4007-1718-5/24/07
https://doi.org/10.1145/3664476.3669928

- **GAN implementation for Network Data:** We present a novel integration of GANs for adversarial resilience within the realm of network security. This represents a significant departure from other systems which were primarily developed for image data. ARGAN-IDS is specifically designed for the unique challenges and complexities of network security. With this in mind, our system takes into account the need to minimize discrepancies and delays in performance as these systems can be deployed in time-constrained security-critical environments.
- **Evaluation of two different GANs:** We evaluate two different approaches based on the training of two different GANs – one with only benign traffic and another with both benign and malicious traffic. This dual approach allowed for a comprehensive evaluation of the trade-offs and nuances of each training method in NIDS. Additionally, we address the difficulty in distinguishing adversarial and malicious traffic, proposing a system that neutralizes the effect of adversarial attacks, maximizing true positives and true negatives, while being cautious not to neutralize actual cyberattacks.
- **Robustness against adversarial attacks:** We improve the robustness of A-NIDS against adversarial attacks. This includes identifying and mitigating potential vulnerabilities in the neural network and developing strategies to improve its resilience while maintaining the accuracy of the neural network in classifying unknown samples, even under adversarial attacks.

2 RELATED WORK

Our ARGAN-IDS system is based on the work proposed for image classifiers, DefenseGAN[18]. DefenseGAN takes advantage of the expressive capabilities of generative models. In their work, a GAN[4] is trained on original data to take out the noise injected through adversarial attacks. In these next sections, we go through a brief overview of Generative Adversarial Networks as well as the work proposed on DefenseGAN as these will be pivotal to understanding our work.

2.1 GANs

A Generative Adversarial Network (GAN)[4] is a sophisticated machine learning model designed to generate synthetic data resembling a given dataset. In a GAN[4], two different models are trained in tandem in an adversarial manner. There's one generative model that emulates the data distribution and a discriminator model that verifies whether or not the given input fits that distribution. The generator receives a random vector z and tries to generate samples that resemble the original ones, while the discriminator network tries to identify which samples are real and which were generated by the generator. These two networks compete and learn from each other in an alternating manner. This process is repeated until the generator can produce samples that are indistinguishable from real samples, and the discriminator cannot distinguish between real and generated samples. After training the generator, it can be used to generate new samples through a compact representation that we call z.

2.2 DefenseGAN

DefenseGAN makes use of the GAN's[4] generator in conjunction with a reconstruction algorithm to neutralize disturbances. In their work, a GAN[4] is trained on original images to take out the noise injected through adversarial attacks. This is achieved through the use of a generator that was trained to learn the distribution of unperturbed images and create data similar to that of the training set. At inference time, they utilize this Generator to find a close output of the original sample by projecting it onto the range of the Generator. The resulting output should, for the most part, be neutralized of adversarial perturbations because the GAN[4] was trained on original samples.

Because the GAN[4] only learns the distribution of a given dataset, it is expected for adversarial examples to be outside the range of function G. Given an input x, and the generator mapping G, by minimizing the reconstruction error $\|G(z) - x\|_2^2$ an $\hat{x}$ can be found that is close to the original example in its manifold and therefore not induce a misclassification. This minimization problem is solved using a set of fixed steps of Gradient Descent and by randomizing the initializations of the vector z. After finding the optimal latent vector $z*$ that minimizes the problem, $G(z*)$ is calculated through the previously trained Generator, resulting in the output with neutralized adversarial perturbations, $\hat{x}$ which is then fed to the classifier.

After applying the DefenseGAN[18] to the original input x, the main classification model receives the neutralized input $\hat{x}$.

3 ARGAN-IDS

Our proposed solution to improve the resilience of NIDS against adversarial attacks harnesses the generative capabilities of Generative Adversarial Networks[4] in an attempt to project an adversarial example back to the region of inputs space that is correctly handled by the network. We assess the applicability of this type of defense on an inherently different machine learning architecture - an anomaly-based NIDS where even a small difference in the reconstructed packet can lead to misclassification. Throughout the work, we verified that the system's resilience was strengthened, and the impact on the detection metrics, when compared to the original model was, in fact, minimal. We deploy this defense on the state-of-the-art anomaly-based NIDS, Kitsune[13].

3.1 Adversarial Attacks on A-NIDS

In the upcoming sections, we analyze the current state of adversarial attacks within the context of Anomaly-Based Network Intrusion Detection Systems (A-NIDS). These systems have shown vulnerabilities against the same adversarial perturbation that affects image classifiers, making the exploration of their weaknesses and potential fortifications crucial.

3.1.1 *Adversarial Attacks on Kitsune's KitNET.* To evaluate the security of our target A-NIDS against adversarial examples, it is necessary to isolate Kitsune's ML component, KitNET, an integral component of the NIDS architecture. In real-world attacks on Kitsune, adversaries must circumvent or overcome the Feature Extractor to introduce perturbations into KitNET's input. However, if adversaries understand the Feature Extractor's workings, they can craft network traffic specifically designed to generate essential features. Hence, the focus of our experiments lies in assessing KitNET's security solely from the perspective of the feature space. KitNET is a lightweight anomaly detection system that employs an ensemble of autoencoders. Its primary output is a root mean square error (RMSE) score, denoted as S, which differs from the probability distributions or logits produced by traditional deep learning classifiers. To trigger an alarm, KitNET utilizes a classification scheme based on the condition: $S \geq \phi\beta$, where ϕ represents the highest recorded value of S during training, and β is a constant that balances the trade-off between false positives and negatives. To employ adversarial attacks on KitNET's autoencoders, a modification is proposed by incorporating a final layer at the output, as represented by Equation 1. This alteration enables the deep learning model to generate classification results based on a threshold value, T. By setting $T = \phi\beta$, the model effectively transforms from an anomaly-based model into a classifier. With this simple modification of KitNET's architecture, it is possible to utilize an adversarial machine learning library developed for image classification models named CleverHans to perform attacks such as the Fast Gradient Sign Method (FGSM).

$$C(x) = \begin{bmatrix} \text{malicious} \\ \text{benign} \end{bmatrix} = \text{RMSE}(x) \begin{bmatrix} 1 \\ -1 \end{bmatrix} + \begin{bmatrix} 0 \\ 2T \end{bmatrix} \tag{1}$$

3.2 Adversarial Defenses on A-NIDS

In the upcoming sections, we navigate through the existing techniques of Adversarial Defenses in Anomaly-based Network Intrusion Detection Systems (A-NIDS). Initially, we confront the inherent limitations of current defense mechanisms, that make distinguishing between genuine and manipulated network traffic a difficult task.

With these limitations in mind, we propose the integration of Generative Adversarial Networks (GANs) as a promising solution to bolster the defenses of NIDS. We go through the potential of GANs in neutralizing adversarial perturbations, discussing their operational adaptability in the face of emerging cyber threats.

3.2.1 Limitations of Current Defenses on A-NIDS. In the context of NIDS, outlier detection-based systems prove to be more practical than classification-based ones due to the time-consuming nature of labeling network traffic and the requirement of expert knowledge for accurate labeling. However, current defenses, such as MagNet[12], fail to differentiate between malicious and adversarial traffic. As a result, Mag-Net's[12] autoencoder reconstruction removes both types of perturbations, leading to the misclassification of all traffic as benign. Thus, alternative defense mechanisms are necessary to overcome this limitation. In the realm of adversarial defense, two common strategies include adversarial training and adversarial detection. Adversarial training involves training the NIDS model with adversarial examples to enhance its robustness. However, this approach is not directly applicable to the current threat model, as NIDS training typically occurs in an unsupervised manner without label information. Adversarial detection, on the other hand, entails training a secondary classifier to identify adversarial examples. Unfortunately, existing detectors trained on benign data struggle to effectively distinguish between adversarial and malicious traffic.

3.2.2 Generative Adversarial Networks as a defense. To address these challenges, the utilization of GANs emerges as a promising solution. GANs, renowned for their generative capabilities and many times outperforming autoencoders, can be leveraged to remove adversarial perturbations on the statistical feature space used by NIDS. By training a GAN to generate realistic network features, the NIDS can compare incoming traffic against these generated features and identify any deviations caused by adversarial attacks. This approach enhances the NIDS's ability to differentiate between genuine and adversarial traffic, reducing the risk of misclassification.The NIDS can then flag such traffic for further analysis or take appropriate defensive measures. On the other hand, it is important to note that GAN-based defenses are not a one-size-fits-all solution and should be tailored to the specific characteristics of the network environment and the types of attacks being targeted. Robust training procedures, proper selection of network architecture, and careful evaluation of performance metrics are crucial for the successful implementation of GAN-based defenses in NIDS.

3.3 ARGAN-IDS architecture

Our layer of defense against AEs is heavily inspired by the works of MagNet[12] and Defense-GAN[18], which were previously applied to the image classification domain, with MagNet taking advantage of an autoencoder to learn data representation and Defense-GAN using GANs. We make use of this idea of a *neutralizer* network that tries to approximate an AE back to the original example's manifold. The logic behind this is that we can leverage the power of generative networks to learn the distribution of the data manifold of original examples. With that knowledge, we can convert AEs into a non-adversarial original example by projecting it onto the latter's manifold, before classification takes place. Because it is essentially an add-on to the neural network, this defense method can be added to any model independently of the architecture, serving as an adversarial attack "neutralizer" before the main model itself.

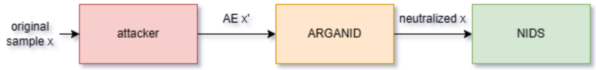

Figure 1: ARGAN-IDS neutralizing inputs provided to A-NIDS

With everything put together, ARGAN-IDS assumes the existence of a GAN acting as a *neutralizer* that receives the input firsthand, before the main model and tries to remove the adversarial perturbation from the AE into an original example before feeding it to the model for classification. When the given input is not an AE, this GAN *neutralizer* should only apply small changes (if any) and therefore it does not have a big effect on the original accuracy. We show this empirically in later sections. Our system integrates GAN in a novel domain - network security. In these systems, both the architecture and the inner workings of the deep neural networks are completely different from traditional classifiers. Thus we assess whether ARGAN-IDS is effective in defending these inherently different networks because in A-NIDS, any slight deviation from the original learned distribution can lead to wrong predictions. Additionally, the data structure and distribution of network packet statistics are conceptually different than those of images. Therefore we also verify if this system can effectively be used to generate valid and realistic samples. Finally, we discuss some limitations related to the integration of GAN in NIDS, which we go over in the next section.

3.3.1 ARGAN-IDS challenges. Our case study makes use of an auto-encoder to perform anomaly-based classification for network traffic. To this extent, whenever a given packet input is given to the model and it differs from the learned representation of normal traffic packets, it is classified as an attack on the network. When using ARGAN-IDS we don't want to lose the expressiveness of the given input packet before giving it to our model for classification; in other words, we want to neutralize the effect of an adversarial attack to maximize true positives and true negatives, but we need to be extra careful to make sure our ARGAN-IDS does not neutralize cyberattacks (since it may raise the amount of false negatives). To prevent this, we need to augment the dataset originally composed of normal network traffic, with malicious packets derived from known network attacks. The tradeoff of this approach is that new unknown attacks can be neutralized by the ARGAN-IDS, to raise the resilience against AAs. To verify this, we trained two different GANs. One GAN was trained using only benign traffic and another one was trained with both benign and malicious traffic. In later sections, we present our findings and the trade-offs with each approach.

4 IMPLEMENTATION

In this section, we go through two main points: the changes conducted on Kitsune to perform adversarial attacks and the GANs implementation and inference reconstruction algorithm. Firstly we explain our prototype to generate and perform adversarial attacks on Kitsune in a white-box attack model. Then, we go on to briefly describe the two different approaches developed in our GAN-based defense and the reconstruction algorithm utilized at inference time.

4.1 Adversarial Attack Generation

Our experiments are centered around Kitsune's NN, KitNET, which primarily uses root mean square error (RMSE) for anomaly detection. This serves as a measure of how much the current packet deviates from what the model considers 'normal' based on a threshold. However, to generate adversarial attacks on this system we had to restructure KitNET to output a vector of probabilities for each class. For the generation of adversarial attacks, we turned to CleverHans[14] – a library originally developed with the image domain in mind. Its primary design is targeted towards machine learning classifiers that process image data. The adversarial attacks generated by CleverHans, especially the Fast Gradient Sign Method (FGSM), are primarily devised to exploit the vulnerabilities of image-based classifiers by adding meticulously crafted perturbations to input images, making them misclassified. Thus, to utilize CleverHans to generate AAs, we modified KitNET to function more like a classifier, using the threshold value of the original system as the classification boundary. Furthermore, given that KitNET was originally implemented using Numpy without any provisions for automatic differentiation, the integration with CleverHans, which demands gradient computations, posed challenges. To bridge this gap, we had to reimplement the inference logic of KitNET using TensorFlow. This transition was imperative, primarily because CleverHans relies heavily on TensorFlow's GradientTape for calculating gradients required for the adversarial attack generation, especially for FGSM. With TensorFlow's GradientTape, it became feasible to compute the necessary gradients for the input with respect to the loss, which is a prerequisite for FGSM.

4.2 ARGAN-IDS implementation

To verify the applicability of our ARGAN-IDS defense system, we developed two different GANs in TensorFlow with different architectures tailored for the two different data distributions and to capture their intricacies. One GAN was trained with only benign traffic, while the other was augmented with both benign and malicious traffic. We verified that the first approach is not well suited for NIDS. In this approach, with the GAN being only trained on benign traffic, it could only generate benign traffic. Thus, when reconstructing the packet's statistics through the GAN, all malicious traffic would be misclassified as benign. We devised a method to try to counteract this limitation based on an arbitrary threshold, presented in Equation 2.

$$\begin{cases} \|G(z) - x\|_2^2 \leq \theta & reconstruct \quad packet \\ else & do \quad nothing \end{cases} \tag{2}$$

That is, only pass the reconstructed packet to the classifier for analysis if the Euclidean distance between the reconstruction of the generator and the original sample is under a certain threshold θ. This method was based on the intuition that adversarial examples would lie closer to the output of the generator than malicious traffic. The tradeoff with this approach is that it would only be able to reduce the false positives introduced by adversarial attacks and not false negatives. This is because it would only be able to reconstruct benign traffic. Through our experiments we found it increasingly hard to set an accurate threshold as adversarial perturbations get more pronounced, making it hard to distinguish between malicious and adversarial traffic. A wrong choice of threshold would result in the introduction of a large amount of false negatives. Thus, throughout the rest of this document, we focus on the second approach where the GAN was trained with both malicious and benign traffic.

4.3 Reconstruction algorithm

Finally, we implemented the reconstruction algorithm. This algorithm aims to find the latent representation $\mathbf{z}^*$ that, when passed through the generator, produces an output that is as close as possible to the input $\mathbf{x}$. This is achieved by minimizing the Euclidean distance between the generator's output and $\mathbf{x}$ in the statistical feature space through L Gradient Descent steps. The algorithm is described in Algorithm 1.

We randomly initialize the latent representation used by the GAN, z, R times because GANs often have complex loss surfaces with many local minima. By initializing z multiple times and running the optimization, we increase the chances of finding a more global (or at least a better local) minimum. Furthermore, we utilize an optimizer with momentum to prevent the algorithm from staling at a local minimum and apply a learning rate scheduler, that starts relatively high and decays over time. This allows for more significant updates initially, which can help escape local minima and smaller updates later on to fine-tune the solution

Algorithm 1 Reconstruction Algorithm for ARGAN-IDS

```
Require: $x$: feature vector
Require: $generator$: GAN's generator model
Require: $L$: Number of gradient descent steps
Require: $R$: Number of random restarts
Initialize $best\_loss$ to $\infty$
Initialize $best\_z$ to null
for $r = 1$ to $R$ do
    Initialize $z$ randomly from a normal distribution
    for $l = 1$ to $L$ do
        Calculate loss: $loss = \|G(z) - x\|^2$
        Compute the gradient of $loss$ w.r.t. $z$
        Update $z$ using the optimizer step
        if $loss < best\_loss$ then
            $best\_loss = loss$
            $best\_z = z$
        end if
    end for
end for
return $best\_z$
```

In summary, this algorithm finds the best latent representation $\mathbf{z}^*$ for a given $\mathbf{x}$ by running gradient descent from multiple random starting points and choosing the one that gives the output from the generator closest to $\mathbf{x}$. This approach is especially useful when trying to manipulate the latent representations of data points in a GAN.

4.4 Parallel optimization

Since we are trying to improve the resilience of NIDS, which can be deployed for real-time detection environments we devise a simple optimization of the previously described algorithm to make the reconstruction step faster. Its goal remains the same: to find the latent vector $\mathbf{z}^*$ that minimizes the reconstruction error $\|G(\mathbf{z})-\mathbf{x}\|_2^2$ using gradient descent with a fixed number of multiple restarts but by parallelizing the R restarts instead. The key to this optimization is in how TensorFlow operations are vectorized. When computing the loss or applying gradients, the function doesn't iterate through each of the R different $\mathbf{z}$ values. Instead, it processes them all simultaneously, as if they were a batch of data. This makes the function significantly faster, especially when using hardware accelerators like GPUs.

To achieve this, the feature vector x is expanded R times to match the number of restarts and ensure we have one separate x for each separate restart r. This means each of the R restarts will simultaneously try to minimize the distance between $G(\mathbf{z})$ and the same $\mathbf{x}$. Similarly, z is randomly initialized with R random vectors. From this point on, this parallelized version behaves similarly to the original: The loss between the generator's output, $G(z)$, (for all R $\mathbf{z}$ values) and the expanded $\mathbf{x}$ is computed, then the gradient of this loss with respect to $\mathbf{z}$ is found and used to update z in the gradient's direction and finally a condition checks if the current loss for each z is better than the previous best loss. If so, the best loss and corresponding z are updated. After all gradient descent steps are done, the function picks the z with the lowest loss among the R restarts and the best latent representation z that minimizes $\|G(\mathbf{z}) - \mathbf{x}\|_2^2$.

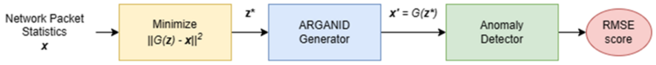

Figure 2: ARGAN-IDS neutralization functions

In summary, this algorithm achieves the same goal as the previous one but does so in a more optimized manner by processing all restarts in parallel. Essentially reducing the time complexity from $O(RL)$ to $O(L)$ where R is the number of random restarts and L is the number of gradient descent steps. This is a clear advantage, especially when R is large, and can significantly speed up the process when using hardware like GPUs that are designed for parallel computations. In our particular case, since we are aiming to protect an IDS, this optimization step is crucial as these systems are often deployed for real-time detection.

5 EVALUATION

To evaluate the performance of the Kitsune Intrusion Detection System (IDS) with and without ARGAN-IDS, we seek to answer five research questions:

- Can IDS maintain its detection results when in tandem with GAN? (Section 5.2)
- Can GANs trained with malicious and benign traffic reduce Adversarial Attack perturbations? (Section 5.3)
- Are GANs equally effective against all perturbation norms ϵ? (Section 5.4)
- Can GANs be used in time-constrained environments? (Section 5.6)
- Can GANs be used to effectively detect AA perturbations? (Section 5.5)

5.1 Experimental Settings

To evaluate the proposed GAN-based defense mechanism against adversarial attacks on Kitsune's KitNET, a series of experiments were conducted under specific settings. This section details the experimental setup, the dataset used, the attack mechanism implemented, and the hardware configuration. The Generative Adversarial Networks (GANs) underwent a training phase to ensure they could effectively generate realistic network features. This training was conducted on 200 epochs for a batch size of 256 and was facilitated using the hardware acceleration capabilities of an Nvidia RTX 3050 graphics card. The training data played an important role in this phase. It encompassed a balanced dataset with 100,000 benign network packets and 100,000 malicious packets, ensuring the GAN learned from both regular and anomalous patterns. Post-training, the inference phase's dynamics were crucial for evaluating the defense mechanism's real-world efficacy. All inference experiments were deliberately run on a CPU to emulate environments where specialized GPU resources might not be readily available. Also, for smaller batch sizes, the time needed to transfer the data to the GPU's memory far outweighs the execution time. The generator's reconstruction capabilities, essential for neutralizing adversarial perturbations, were dictated by two primary parameters: gradient descent steps and noise vector initializations. With 200 gradient descent steps, the generator iteratively minimized the reconstruction error, while the noise vector's 10 random initializations ensured diverse synthetic data generation. To challenge the defense mechanism, a renowned adversarial attack technique, the Fast Gradient Sign Method (FGSM), was employed through the CleverHans library. For this research, the Mirai attack dataset was selected. This dataset aligns with the data used by Kitsune, ensuring the experimental results are directly comparable and relevant to real-world NIDS deployments. This dataset contains communications established between an ICT organization and the internet. We make use of the Mirai attack dataset's labels, to distinguish between malicious communications (i.e., all communications established to the Mirai command and control) and benign (i.e., all communcations established to the other hosts). All these experiments were done on a Ryzen 7 5800X CPU, complemented by 16 GB of RAM and an Nvidia RTX 3050 graphics card. Notably, the system operated on Windows 11, running the Windows Subsystem for Linux (WSL).

5.2 ARGAN-IDS generalization

To determine whether or not GANs can generalize and generate data for the context of an IDS, we compare the baseline performance of Kitsune IDS to its performance when integrated with ARGAN-IDS (without adversarial attacks).

5.2.1 Baseline Kitsune IDS. The baseline Kitsune IDS has a high accuracy of approximately 91.74%. The precision is remarkably high at 99.98%, indicating that almost all the traffic it labeled as malicious truly was malicious. The recall, on the other hand, is at 84.73%, suggesting that Kitsune IDS missed a certain portion of actual malicious traffic. The F1-score, which combines precision and recall, is also high at 91.72%.

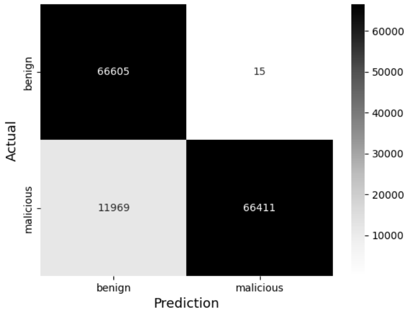

Figure 3: Confusion matrix of baseline Kitsune

Table 1: Kitsune baseline metrics

Accuracy	Precision	Recall	F1-score
0.91735172413	0.99977418480	0.84729522837	0.91724099830

5.2.2 Kitsune IDS with ARGAN-IDS. With the ARGAN-IDS integration, the accuracy drops slightly to 90.64% (compared to the baseline), the precision remains extremely high at 99.99%. Recall, however, decreases to 82.70% suggesting ARGAN-IDS introduces a small amount of false negatives. The F1-score, indicative of the balance between precision and recall, is 90.53%. While there is a slight reduction in accuracy and F1-score when the Kitsune IDS is integrated with the ARGAN-IDS, it's crucial to note that the drop is minimal. Given the added protection against adversarial attacks, this minor reduction in performance can be considered a trade-off for enhanced security and robustness against adversarial threats.

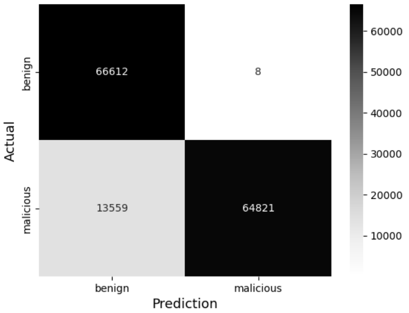

Figure 4: Confusion matrix of Kitsune when integrated with ARGAN-IDS

Table 2: Kitsune + ARGAN-IDS metrics

Accuracy	Precision	Recall	F1-score
0.90643448275	0.99987659843	0.82700944118	0.90526433394

5.3 Adversarial attack robustness

To assess whether or not GANs can be effectively used to reduce adversarial perturbations in the context of an IDS, we look at the performance of the Kitsune IDS under adversarial attacks and compare it to its performance when protected by the ARGAN-IDS. For this experiment we used FGSM as the adversarial attack algorithm, using 3.5 as the perturbation norm. This was the minimal perturbation that introduced a misclassification.

5.3.1 Without ARGAN-IDS. Under adversarial attacks, the Kitsune IDS's efficacy is heavily compromised. The system entirely misses the detection of true negatives (TN = 0), leading to a large number of false positives (FP = 66620). This essentially means that the benign traffic is mostly misclassified as malicious. The accuracy drops to approximately 54.06%, and the F1-score is at 70.18%. However, recall is at 100% because all detected anomalies are classified as true positives (since no benign traffic was detected). This scenario demonstrates the vulnerability of Kitsune IDS to adversarial attacks.

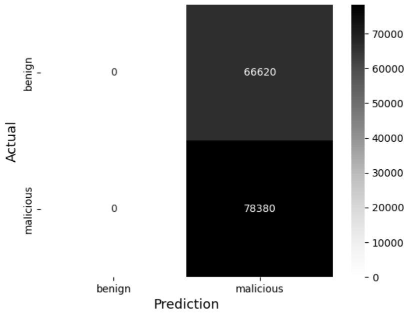

Figure 5: Confusion Matrix of Kitsune when attacked with adversarial perturbation $\epsilon = 3.5$

Table 3: Kitsune with adversarial attacks metrics

Accuracy	Precision	Recall	F1-score
0.54055172413	0.54055172413	1.0	0.70176381054

5.3.2 With ARGAN-IDS. When protected by the ARGAN-IDS, the Kitsune IDS's efficacy shows notable improvement under adversarial conditions. The true negatives increase dramatically (TN = 66134), and the false positives drop to a minimal 486. The accuracy increases 36.75% to approximately 90.80%, remaining at a high value when compared to the baseline. Precision is at 99.26%, which, although slightly reduced from the baseline, is still very high. Recall is at 83.61%, which is comparable to the baseline. The F1-score is 90.77%, which is also comparable to the ARGAN-IDS. These metrics suggest that ARGAN-IDS effectively mitigates the impact of adversarial attacks, allowing Kitsune IDS to maintain high efficacy.

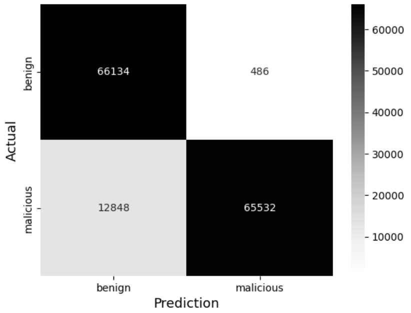

Figure 6: Confusion Matrix of Kitsune when integrated with the ARGAN-IDS for adversarial perturbation $\epsilon = 3.5$

Table 4: Kitsune and ARGAN-IDS metrics for adversarial perturbation $\epsilon = 3, 5$

Accuracy	Precision	Recall	F1-score
0.90804137931	0.99263837135	0.83608063281	0.90765800080

5.4 Effect of adversarial noise perturbation ϵ

The resilience of an Intrusion Detection System (IDS) like Kitsune, when integrated with ARGAN-IDS, can be further elucidated by examining how it performs under varying degrees of adversarial noise perturbations. This section investigates the system's robustness specifically when the adversarial perturbation ϵ is intensified to 10, a substantial increase from the previous scenario where ϵ equaled 3.5.

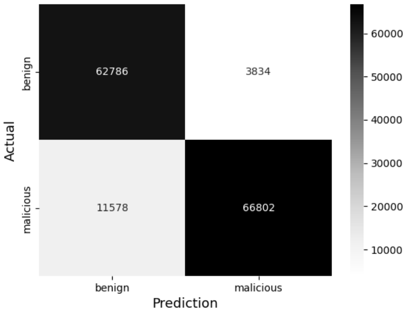

Figure 7: Confusion Matrix of Kitsune when integrated with the ARGAN-IDS for adversarial perturbation $\epsilon = 10$

Table 5: Kitsune and ARGAN-IDS metrics for adversarial perturbation $\epsilon = 10$

Accuracy	Precision	Recall	F1-score
0.89371034482	0.94572172829	0.85228374585	0.89657486444

The depicted results indicate a noticeable shift in performance metrics. With an adversarial perturbation of $\epsilon = 10$, the system maintains a commendable accuracy rate of approximately 89.37%. This statistic, while slightly lower than previous measurements under less intense adversarial conditions, signifies the system's robustness while under more aggressive attacks. Precision experiences a noticeable decline to 94.57%. This reduction suggests that as the adversarial noise increases, the system faces challenges in distinguishing between normal and malicious traffic, resulting in a higher rate of false positives. Despite this, the precision metric at this level of perturbation remains relatively high, highlighting the system's ability to correctly identify a large proportion of malicious instances. Recall, on the other hand, is at 85.23%, which indicates that the system, even under intensified adversarial conditions, successfully identifies a significant majority of the actual malicious activities. These results confirm that while Kitsune integrated with ARGAN-IDS exhibits some susceptibility to high degrees of adversarial noise, its overall robustness in such hostile environments remains largely uncompromised. The system continues to perform at a high standard, particularly in terms of maintaining substantial accuracy and recall metrics, despite the elevated level of adversarial noise.

It is worth noting that although it may seem that using a bigger perturbation norm might be desirable for the attacker, this is not the case. Bigger ϵ leads to an increasingly obvious discrepancy between original and adversarial samples, thus making such attacks easily detected. Thus, the attacker needs to devise strategies that can induce misclassifications through a small enough ϵ to avoid detection. In the next section, we show how ARGAN-IDS can also be used for this aforementioned detection.

5.5 Adversarial attack detection

Since our ARGAN-IDS was trained with unperturbed samples from benign and malicious traffic, we expect both these data distributions to lie closer to the range learned by the ARGAN-IDS generator than adversarial examples. Thus, in theory, the Euclidean distance that the ARGAN-IDS tries to minimize when reconstructing a sample will be larger in value for adversarial examples. With this in mind, by setting a threshold and comparing it against the Mean Squared Error (MSE) of the original sample and its reconstruction, we can effectively identify whether or not the given input is adversarial or not. Formally, for an arbitrarily chosen threshold θ:

$$\begin{cases} \|G(z) - x\|_2^2 \geq \theta & adversarial \quad attack \\ else & no \quad attack \end{cases} \tag{3}$$

We calculated the MSE score for the first 200000 samples for both clean and perturbed data. The results are concatenated in the following confusion matrix.

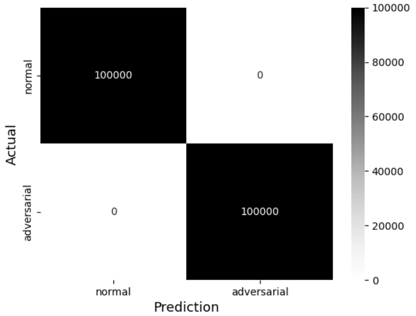

Figure 8: Confusion matrix of ARGAN-IDS when used for adversarial detection

Table 6: ARGAN-IDS when used for adversarial detection metrics

Accuracy	Precision	Recall	F1-score
1.0	1.0	1.0	1.0

These results show that this attack detection strategy can be exceptionally effective given our parametrization, being able to distinguish between normal and adversarial samples with a 100% success rate. It is worth noting that this approach does not verify whether or not the adversarial attack was successful in producing a misclassification of the model. Instead, it focuses on the introduced perturbations, detecting the attack before it even reaches the classification model. In our experiments, even when the FGSM algorithm wasn't able to introduce misclassifications on malicious traffic, the perturbations were still introduced and therefore detected by the ARGAN-IDS.

5.6 ARGAN-IDS execution time

The time it takes for the ARGAN-IDS to reconstruct a given feature vector is highly dependent on the number of Gradient Descent (GD) steps performed to estimate $z*$. The corresponding time complexity is therefore derived from the number of chosen Gradient Descent steps multiplied by the time required to compute those gradients. On the first iteration of our reconstruction algorithm, the number of random restarts also had an impact on time complexity as the Gradient Descent steps were executed sequentially on each random restart R. With the revised parallelized version, the number of random restarts has practically no effect on running time, since the GD algorithm is performed in parallel in each random restart. We found that the average time required for the ARGAN-IDS to find the reconstruction for each given packet to be around the order of 0.7 seconds, with the unoptimized version performing up to R times slower. Usually, for most applications, these running times would not pose a problem. In the case of network intrusion detection systems, with the huge amount of packets that have to be processed and the need for real-time detection, this delay can introduce a limitation. Nevertheless, it is worth noting that different choices of the number of Gradient Descent steps L present a tradeoff between running time and adversarial attack robustness as well as the system's overall accuracy.

5.7 ARGAN-IDS trained with benign traffic

One of the benefits of A-NIDS is the ability of these systems to detect new cyberattacks that may emerge. This is possible because an A-NIDS is only trained on benign traffic. Therefore, it would be ideal if, in the same way, ARGAN-IDS was trained only on benign traffic. With this approach, however, the GAN is only capable of generating benign traffic, resulting in the misclassification of all malicious traffic, artificially increasing the False Negatives. To combat this, we proposed to only reconstruct the input sample if the GAN's reconstruction error is small, thus avoiding the neutralization of malicious traffic. This solution is based on the assumption that adversarial attacks are closer to benign traffic than malicious traffic. However, this solution presents a tradeoff in that, in this case, the GAN would only be able to reduce false positives introduced by adversarial attacks since it would not be capable of reconstructing malicious traffic. Throughout our experiments, we observed that adversarial perturbations introduced in benign traffic do indeed lie closer to the original benign traffic than malicious traffic. However, this behavior only holds for small perturbation norms and it gets increasingly difficult to distinguish between adversarial and malicious traffic as the perturbation norm gets bigger. To sum up, training the GAN with only benign traffic presents a few limitations as it doesn't protect against adversarial attacks on malicious traffic and presents a possible vulnerability for high adversarial perturbation norms. We present the detection results of the IDS in a normal execution and under adversarial attacks for $\epsilon = 3.5$ using this defensive approach. Adversarial perturbations with a norm higher than 3.5 are increasingly difficult to distinguish from malicious traffic. To provide adversarial resilience for higher perturbation norms, the end-user would have to increase the threshold where the reconstruction takes place. This increase in adversarial resilience would come at the cost of added False Negatives with an increased risk of malicious traffic being neutralized by the GAN. However, it is important to note that adversarial attacks desirably introduce small perturbations, as bigger perturbation norms are often trivial to detect by a human observer. On the other hand, as shown in Table 7, the impact of these low norm attacks can be significant, introducing high amounts of misclassifications with up to 100% success rate.

Table 7: Success rate and misclassifications introduced by different adversarial perturbation norms

ϵ	1.5	2	2.5	3	3.5
Missclassifications	16	17	21	52156	65000
Success Rate	0.02462%	0.02615%	0.0323%	80.24%	100%

5.7.1 *ARGAN-IDS trained with benign traffic in a normal execution.*

When integrated with ARGAN-IDS trained on benign data, Kitsune IDS maintains similar accuracy and precision levels as the baseline. The slight decrease in recall and F1-score is minimal, indicating that the integration of ARGAN-IDS does not significantly affect the system's ability to detect true malicious traffic under normal conditions.

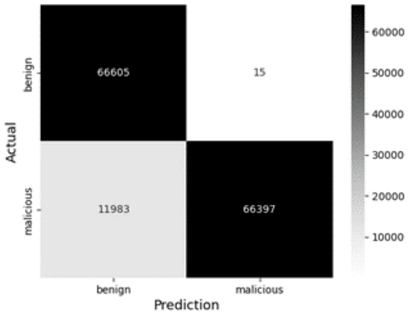

Figure 9: Confusion Matrix of Kitsune when integrated with the ARGAN-IDS (trained with benign traffic)

Table 8: Kitsune and ARGAN-IDS (trained with benign traffic) metrics for adversarial perturbation $\epsilon = 3.5$

Accuracy	Precision	Recall	F1-score
0.91725517241	0.99977413720	0.84711661138	0.91713630587

5.7.2 *ARGAN-IDS trained with benign traffic in execution with adversarial attacks.*

As shown previously, under adversarial attack conditions with a perturbation norm of $\epsilon = 3.5$, the performance of Kitsune IDS is significantly compromised. In contrast, when Kitsune IDS is integrated with ARGAN-IDS trained with benign traffic and subjected to the same adversarial conditions, the system maintains high accuracy and precision. There's a notable improvement in recall, indicating better detection of anomalies under adversarial attack conditions. This suggests that ARGAN-IDS provides a significant layer of protection against adversarial perturbations.

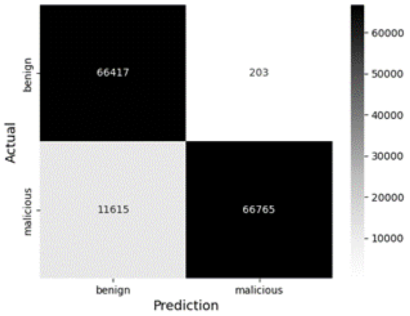

Figure 10: Confusion Matrix of Kitsune when integrated with the ARGAN-IDS (trained with benign traffic) for adversarial perturbation $\epsilon = 3.5$

Table 9: Kitsune and ARGAN-IDS (trained with benign traffic) metrics for adversarial perturbation $\epsilon = 3.5$

Accuracy	Precision	Recall	F1-score
0.91849655172	0.99696870146	0.85181168665	0.91869169166

5.8 ARGAN-IDS training comparison

When a Generative Adversarial Network (GAN) is trained using both benign and malicious data, it gains a significant capacity to reconstruct and neutralize adversarial attacks in benign and malicious network traffic. This training approach allows the GAN to address adversarial perturbations effectively across various perturbation norms ϵ. While it becomes increasingly challenging to neutralize these perturbations as the norm grows, the GAN's ability to detect and halt adversarial attacks enhances its overall resilience. This detection capability complements the GAN's resilience, with the ease of detecting adversarial attacks growing alongside the perturbation norms.

In contrast, a GAN trained solely on benign data demonstrates limitations. It can only reconstruct and neutralize adversarial attacks on benign traffic, reducing false positives (FP) introduced by such attacks. The resilience added in this case depends on a set threshold, a balance between resilience and the introduction of false negatives (FN) in the Intrusion Detection System (IDS). A higher threshold might inadvertently cause the GAN to classify malicious traffic as benign. As perturbation norms increase, distinguishing them from malicious traffic becomes more difficult, posing a vulnerability for high ϵ values.

In conclusion, while GANs offer a promising approach to enhancing network security against adversarial perturbations, their efficacy depends heavily on the nature of the training data and the specific parameters set for detection and resilience. The balance between false positives and negatives, along with the GAN's ability to adapt to varying perturbation norms, remains crucial in its application to network intrusion detection systems.

6 CONCLUSIONS

In this work, the integration of Generative Adversarial Networks (GANs), specifically ARGAN-IDS, with the Kitsune Intrusion Detection System (IDS) was carefully evaluated. Our experiments aimed to answer several important questions, primarily focusing on the GAN's efficacy in neutralizing adversarial perturbations and its adaptability in real-time environments. From our experiments, the baseline performance of Kitsune IDS already showcased commendable results, with an accuracy of around 91.74%. However, when confronted with adversarial attacks, the vulnerability of the system became evident, with accuracy plummeting to around 54.06%. This reduction in performance underlines the critical need for effective adversarial defenses. Introducing ARGAN-IDS into the IDS environment yielded promising results. While there was a very slight reduction in the IDS's overall detection metrics, the system's resilience against adversarial attacks was significantly improved. With ARGAN-IDS, the accuracy under adversarial conditions rebounded to approximately 90.80%, demonstrating the GAN's potential in effectively mitigating adversarial perturbations. Furthermore, ARGAN-IDS showcased its capability to detect adversarial perturbations beforehand, distinguishing between normal and adversarial samples with a 100% success rate. In summary, the use of GANs in tandem with IDSs presents a promising strategy for improving the robustness of cyber defense mechanisms. The ARGAN-IDS, with its capability to neutralize adversarial perturbations, offers a robust shield against adversarial threats. However, there's a pressing need to address the associated time complexities to ensure real-time applicability as well as the limitations that training with malicious training assumes.

ACKNOWLEDGEMENT

Work supported by the European Commission through contract 101021797 [1].

REFERENCES

[1] Huda Ali Alatwi and Amjad Aldweesh. 2021. Adversarial Black-Box Attacks Against Network Intrusion Detection Systems: A Survey. In *2021 IEEE World AI IoT Congress (AIIoT)*. IEEE, 0034–0040.

[2] Qiumei Cheng, Shiying Zhou, Yi Shen, Dezhang Kong, and Chunming Wu. 2021. Packet-level adversarial network traffic crafting using sequence generative adversarial networks. *arXiv preprint arXiv:2103.04794* (2021).

[3] Joseph Clements, Yuzhe Yang, Ankur A Sharma, Hongxin Hu, and Yingjie Lao. 2021. Rallying adversarial techniques against deep learning for network security. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 01–08.

[4] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. 2018. Generative adversarial networks: An overview. *IEEE signal processing magazine* 35, 1 (2018), 53–65.

[5] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[6] Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. 2017. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280* (2017).

[7] Dongqi Han, Zhiliang Wang, Ying Zhong, Wenqi Chen, Jiahai Yang, Shuqiang Lu, Xingang Shi, and Xia Yin. 2021. Evaluating and improving adversarial robustness of machine learning-based network intrusion detectors. *IEEE Journal on Selected Areas in Communications* 39, 8 (2021), 2632–2647.

[8] Ke He, Dan Dongseong Kim, Jing Sun, Jeong Do Yoo, Young Hun Lee, and Huy Kang Kim. 2022. Liuer Mihou: A Practical Framework for Generating and Evaluating Grey-box Adversarial Attacks against NIDS. *arXiv preprint arXiv:2204.06113* (2022).

[9] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236* (2016).

[10] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. 2018. Adversarial examples in the physical world. In *Artificial intelligence safety and security*. Chapman and Hall/CRC, 99–112.

[11] Jiajun Lu, Theerasit Issaranon, and David Forsyth. 2017. Safetynet: Detecting and rejecting adversarial examples robustly. In *Proceedings of the IEEE international conference on computer vision*. 446–454.

[12] Dongyu Meng and Hao Chen. 2017. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*. 135–147.

[13] Yisroel Mirsky, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai. 2018. Kitsune: an ensemble of autoencoders for online network intrusion detection. *arXiv preprint arXiv:1802.09089* (2018).

[14] Nicolas Papernot, Fartash Faghri, Nicholas Carlini, Ian Goodfellow, Reuben Feinman, Alexey Kurakin, Cihang Xie, Yash Sharma, Tom Brown, Aurko Roy, et al. 2016. Technical report on the cleverhans v2. 1.0 adversarial examples library. *arXiv preprint arXiv:1610.00768* (2016).

[15] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. 2016. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, 372–387.

[16] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. 2016. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*. IEEE, 582–597.

[17] Han Qiu, Tian Dong, Tianwei Zhang, Jialiang Lu, Gerard Memmi, and Meikang Qiu. 2020. Adversarial attacks against network intrusion detection in iot systems. *IEEE Internet of Things Journal* 8, 13 (2020), 10327–10335.

[18] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. 2018. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605* (2018).

[19] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).

[20] Martin Teuffenbach, Ewa Piatkowska, and Paul Smith. 2020. Subverting Network Intrusion Detection: Crafting Adversarial Examples Accounting for Domain-Specific Constraints. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*. Springer, 301–320.

[21] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. 2017. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204* (2017).

[22] Jianyu Wang, Jianli Pan, Ismail AlQerm, and Yuanni Liu. 2021. Def-ids: An ensemble defense mechanism against adversarial attacks for deep learning-based network intrusion detection. In *2021 International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 1–9.

[23] Han Xu, Yao Ma, Hao-Chen Liu, Debayan Deb, Hui Liu, Ji-Liang Tang, and Anil K Jain. 2020. Adversarial attacks and defenses in images, graphs and text: A review. *International Journal of Automation and Computing* 17, 2 (2020), 151–178.

[24] Jiliang Zhang and Chen Li. 2019. Adversarial examples: Opportunities and challenges. *IEEE transactions on neural networks and learning systems* 31, 7 (2019), 2578–2593.

[25] Bolor-Erdene Zolbayar, Ryan Sheatsley, Patrick McDaniel, Michael J Weisman, Sencun Zhu, Shitong Zhu, and Srikanth Krishnamurthy. 2022. Generating Practical Adversarial Network Traffic Flows Using NIDSGAN. *arXiv preprint arXiv:2203.06694* (2022).

[1] H2020 STARLIGHT https://www.starlight-h2020.eu/