An Approach to Improve the Robustness of Machine Learning based Intrusion Detection System Models Against the Carlini-Wagner Attack

Medha Pujari
EECS Department
University of Toledo
Toledo, USA
medharani.pujari@rockets.utoledo.edu

Bhanu Prakash Cherukuri
EECS Department
University of Toledo
Toledo, USA
bcheruk@rockets.utoledo.edu

Ahmad Y Javaid
EECS Department
University of Toledo
Toledo, USA
ahmad.javaid@utoledo.edu

Weiqing Sun
ET Department
University of Toledo
Toledo, USA
weiqing.sun@utoledo.edu

***Abstract*—Machine Learning (ML) techniques have been applied over the past two decades to improve the abilities of Intrusion Detection Systems (IDSs). Over time, several enhancements have been implemented to help the ML-based IDS models tackle the ever-evolving attack behaviors. However, recent works reveal that ML models are vulnerable to adversarial perturbations. With the increasing volumes of data passing through systems, defeating adversarial attacks has become a significant challenge. Recent research suggests that Generative Adversarial Networks (GANs) possess a good potential in creating adversarial samples and tackling them, playing well on both offense and defense teams. With a motive to improve the resistance of ML-based IDS models against a powerful white-box evasion attack technique, the Carlini-Wagner, we propose a GAN-based defensive approach and evaluate it with the CSE-CIC-IDS2018 dataset. The paper presents preliminary evaluation results and discusses the direction in which we want to continue the work.**

***Index Terms*—Adversarial Machine Learning, Intrusion Detection Systems, Generative Adversarial Networks, Carlini-Wagner attack**

I. Introduction

A critical task in cybersecurity is to detect network anomalies, for which an Intrusion Detection System (IDS) is an effective solution. In terms of processing time, scalability, and reliability, Machine Learning (ML) approaches have made remarkable progress in almost all the domains the Artificial Intelligence (AI) has conquered. However, unlike in domains like image or natural language processing, progress achieved through ML techniques in the cybersecurity field is relatively lesser. Although creating a fully automated and intelligent cyber defense system is still a long way to go, first-level operators in Network and Security Operation Centers may effectively benefit from ML-based detection and analysis technologies [1]. As one of the defenders in the cybersecurity domain, IDSs become more capable and efficient when powered by ML techniques. Unfortunately, the research behavior in most experiments shows that the main focus in an ML algorithm-design process is on the accuracy with which a model performs, rather than on its resilience and robustness. Almost two decades ago, the experiments revealed the vulnerabilities of the Neural Networks (NNs) to adversarial environments. Eventually, it has become more evident that ML classifiers are vulnerable to well-crafted adversarial examples [2]. Experiments suggest that such deceptive examples result in a significant drop in performance, thereby allowing diverse attack models to defeat ML-based network security applications. Our main motive behind this work is to defend the defenders (i.e., the IDSs) by making them more resistant to adversarial environments wherein the adversaries have almost complete knowledge of their target models.

An adversarial input is a data point with some non-random perturbation introduced into it with careful computation. The perturbation is imperceptible to humans but makes enough difference to an NN-based model to make an incorrect prediction. The field that studies these types of attacks is called Adversarial Machine Learning (AML) and is widely investigated in some areas related to image classification. However, its exploration in cybersecurity-related areas, such as intrusion detection, is relatively shallow [3]. The adversarial samples are typically designed in a way to evade detection [4]. An AML attack can take place in either training or testing phases, or both. An AML attack during the training phase is often called a Poisoning attack, and one which takes place during the testing phase is called an Evasion attack.

Generative Adversarial Networks (GANs), as showcased in recent research works [5], possess a promising nature in handling adversarial environments. Our motive behind choosing the Carlini-Wagner attack - which has been considered a powerful adversarial attack algorithm in the research community - is its complexity and efficiency. Furthermore, a dataset is a fundamental element with a significant contribution to an ML-based model's performance. Therefore, our choice for this effort is the CSE-CIC-IDS2018, a publicly available contemporary IDS dataset with impressive characteristics, as discussed later in this paper.

The significant contribution of this research is to propose and examine the efficacy of a GAN-based defensive mechanism in resisting the Carlini-Wagner attack, using the CSE-CIC-IDS2018 as it represents modern network and attack

978-1-6654-9952-1/22/$31.00 ©2022 IEEE

behaviors. We assess the vulnerability of the dataset against the Carlini-Wagner attack. The defense is evaluated using four efficient classification algorithms.

The remaining portion of this paper is organized as follows - Section II provides a brief background on the elements used/focused upon in our experimentation. Section III covers the research available in the literature related to this work. Section IV discusses the architecture of the experimental framework and its workflow. Section V covers the implementation and evaluation processes. Section VI analyzes the evaluation results and briefly weighs up the work in comparison with related work. Section VII concludes the work and presents our future research direction.

II. Background

A. Dataset Overview

The behavioral patterns of network traffic evolve, requiring the datasets to reflect contemporary characteristics. Unfortunately, most publicly available IDS datasets do not satisfy the current technological demands and lack modern traffic behaviors. Therefore, the shortage of optimal IDS datasets should be addressed, and there is active research progressing in this area [6]. Some essential parameters that determine the quality of an IDS dataset are - the framework used for generating the traffic scenarios; diversity of attack scenarios captured; anonymity; the variety of protocols included; capturing complete network interactions; configurations in the network; feature set; labeled data samples; heterogeneity; and metadata [7]. We have chosen the CSE-CIC-IDS2018 dataset for this effort, considering all these characteristics.

The CSE-CIC-IDS2018 dataset was developed in collaboration with the Communications Security Establishment (CSE) and the Canadian Institute of Cybersecurity (CIC) to generate a dataset in a close-to-realistic network environment. The dataset has over 80 features and covers seven different modern attack scenarios. It has about 83% benign and 17% malign traffic instances. Although the dataset is not well-balanced [8], we chose it because of the diversity in the traffic data behaviors it has and the modern network characteristics represented in it. The dataset has multiple classes - seven of them representing attacks, and one is for normal traffic.

B. The Carlini-Wagner Attack

Carlini and Wagner [9] proposed a targeted evasion attack, hereafter referred to as CW attack, to counter defensive distillation, a popular defense mechanism. This white-box attack became more potent than many other white-box attack techniques in the research community, rendering most defensive methodologies ineffective. Formulating an optimization problem to produce misclassification is the foundation of an adversarial attack algorithm. For CW attack, the problem of creating adversarial examples is represented as follows:

$$\begin{aligned} minimize\ \ d(x, x+\eta) \\ such\ that\ \ Y(x+\eta) = T\ \ (this\ is\ constraint\ 1) \\ where\ \ x+\eta\ \epsilon\ [0,1]^n\ \ (this\ is\ constraint\ 2) \end{aligned} \tag{1}$$

In equation 1, x is an input data point, η is the perturbation, d is the metric of the distance between a real input and its corresponding adversarial form, Y is the classification function, T is the target class chosen by the adversary, and n is the number of dimensions in the feature space. The constraint-1 ensures the misclassification of the data point, while constraint-2 ensures the adversarial sample generated is within the normalized boundaries of the dataset [10] [11].

The authors define the objective function in seven different ways and choose the optimal one, based on which is the closest to the target-class misclassification. The distance metrics are specified using Lp norms (i.e., L0, L2, L∞) [11].

C. Generative Adversarial Network

A GAN is a Deep Learning (DL)-based generative model that was first described by Ian Goodfellow et al. [12]. Its goal is to build adversarial samples, very similar to original data, from an input dataset. A GAN is implemented by two neural networks that challenge each other, as in a two-player game. It attempts to mimic a data distribution and allows a model to learn more from available data.

A random number (random noise) is given as input to the Generator, which generates the samples that are comparable to those in the dataset and forwards them to the Discriminator, which examines the samples and predicts whether they are original data or generated ones [13]. The Discriminator learns the original data characteristics and, based on this knowledge, makes decisions on the data passed to it by the Generator. In simple terms, the Generator keeps improving its adversarial samples to make it difficult for the Discriminator to identify them. The Discriminator tries to learn more and correctly identify the adversarial samples introduced by the Generator. There are some limitations to GAN's capacities. Its *Vanishing Gradient Problem* is one of them, where the Generator reaches a saturation point and can no longer produce new samples to trick the Discriminator. Meaning, the Discriminator identifies the samples created by Generator with high confidence values leaving no gradient for the Generator [14]. This issue can be averted by carefully balancing the race/training between the Generator and Discriminator networks and making sure the Discriminator is not over-trained. Ensuring that the Discriminator is trained to an optimal level for every iteration of Generator training is critical in avoiding such issues. Various approaches for GAN training stabilization are discussed and analyzed in [15]

Using GAN within its capable boundaries, we consider two conventional loss metrics in this work - Generator loss and Discriminator loss. In the following sections, we extend the discussion on how we use GAN and its characteristics for our experiment.

D. Classification Algorithms

Classification algorithms play a vital role in the performance of an ML-based model. Therefore, the algorithms selected for this work have been chosen based on the competence they have projected in the research community, especially in handling datasets that have multiple classes [8].

1) Decision Trees: The primary ability of the Decision Tree (DT) algorithm lies in effectively classifying data into branch-like structures with nodes, leaf nodes, and a root node, which together form a logical tree. This technique is efficient in handling extensive data; therefore, it is a good choice for a dataset like CSE-CIC-IDS2018 [16].

2) Random Forest: Random Forest (RF) classifier is a well-known powerful ensemble learning mechanism that comprises a large number of DTs, which are trained on different portions of a given dataset. Additionally, the features considered for making decisions in each tree also differ depending on varying data patterns. Finally, the predictions of all individual trees are considered and averaged to arrive at the final decision on an input [17].

3) Support Vector Machines: Support Vector Machine (SVM) is a classical algorithm that effectively handles big data applications. It is yet another appropriate technique to handle a dataset like CSE-CIC-IDS2018. Although this algorithm involves complex and heavy computations, we have chosen it considering its efficiency in addressing non-linear classification problems [18]. For this work, linear SVMs are implemented.

III. Literature Review

ML models based on neural networks are vulnerable to adversarial instances, as initially revealed by the experiments conducted by Szegedy et al. [4]. The necessity to investigate various approaches of launching adversarial attacks and developing countermeasures has grown ever since the revelation. Different strategies for generating adversarial samples have been widely studied. Additionally, there are studies on how models perform in adversarial environments with a key focus on the characteristics of contemporary datasets [19].

Ferdowsi et al. [20] propose a distributed GAN to detect abnormal activity in IoT devices. Msika et al. [13] propose an approach called SIGMA where they use GAN and meta-heuristics to improve the resilience of ML-based IDSs. Initially, they train a GAN to generate attack samples using the IDS as a discriminator. This process iterates until the detection system's score does not change for three consecutive rounds. At this point, they generate alternative samples that the GAN might have missed by running a search-based method. After that, the IDS will be trained on the instances generated from the two algorithms and the original dataset. This approach prevents over-fitting by exposing the classifier to real data and generated attacks, improving IDS performance.

Shahriar et al. [21] proposed a GAN-based IDS (G-IDS) that can generate more training data to tackle the problem of imbalanced or missing instances. G-IDS framework is divided into four segments - a database module consisting of real-world data and samples generated by the GAN; an IDS module trained twice - with and without the pending data - to evaluate hybrid data; a controller module which decides whether to accept or improve data; and a data-synthesizer module having GAN as the core component, to generate new samples. The evaluation presented in the paper suggests that the G-IDS framework results in better accuracy than those obtained from the independent IDS studied.

Hara et al. [22] present an approach to design IDS models with intrinsic robustness against adversarial attacks. They employ semi-supervised learning, along with an Adversarial Autoencoder (AAE). An AAE essentially uses the properties of an autoencoder combined with the adversarial loss concept of a GAN. A GAN assists in regularizing key features while an autoencoder decreases the dimensions of input data by extracting key features. Then, latent spaces, z1 (information about ”normal” or ”attack”), and z2 (all other data characteristics) are formed. One of the discriminators applies a categorical distribution to the latent class variable vector, z1. The other one applies a Gaussian distribution to z2. The semi-supervised AAE is trained in three phases using the SGD approach (reconstruction, regularization, semi-supervised classification).

IV. Architecture and Workflow of the Proposed Approach

The experiment comprises a series of independent evaluations. At first, the performance of the baseline model is evaluated in non-adversarial settings, i.e., with the original dataset, using each of the chosen classification algorithms. Next, the model is evaluated by implementing the CW attack. Later come the incorporation of the defense layer into the model and its performance evaluation.

Figures 1 and 2 outline the training and testing phases, respectively. The reminder of this section discusses classification behaviors and technical specifications involved in the experiment, and briefly describes the components that constitute the architecture.

A. Classification Goal

The following are the classification goals of the experimental setup:

- the discriminator is to classify adversarial and non-adversarial data, therefore, this is binary classification.
- the final IDS is to classify whether the data instances it receives are benign or malign, which is a binary classification, too.

B. Hardware and Software Specifications Used for the Experiments

The computer that was used for the experiment had the following specifications: Intel Xeon Processor E5-2697 @ 2.6 GHz, 128 GB RAM, and Microsoft Windows 64-bit. The software specifications include Python 3.6.5, Scikit-learn 0.24.2, Tensorflow 1.13.2, and Keras 2.1.5. The CW attack was implemented using the Cleverhans 3.0.1 library [23].

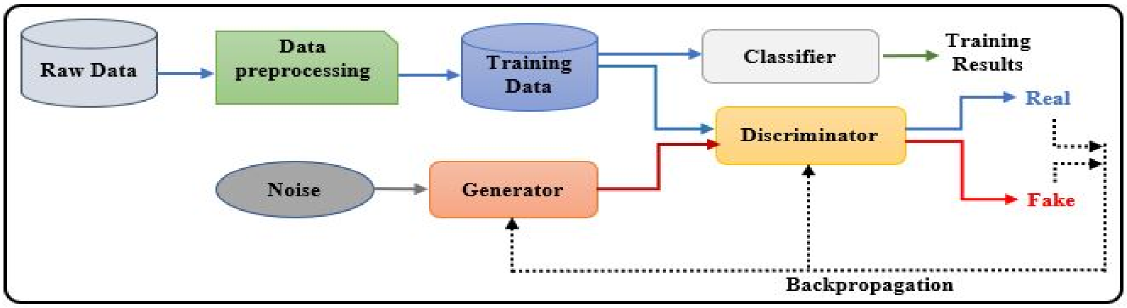

Fig. 1. Training phase of the proposed GAN-based framework.

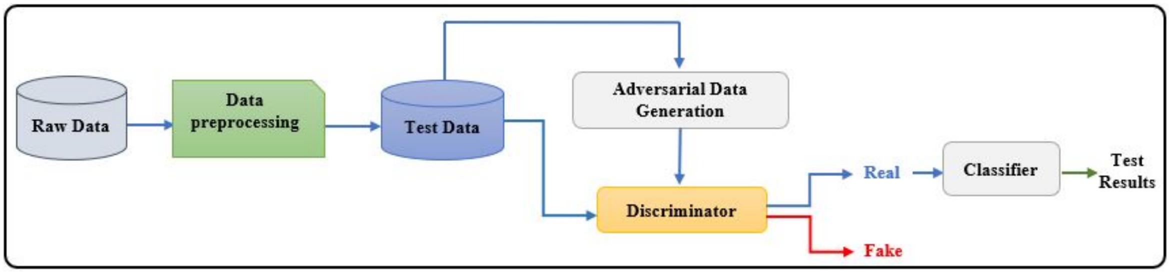

Fig. 2. Testing phase of the proposed GAN-based framework.

C. Configuration of the Components Involved

1) Generator Configuration: The Generator is made up of a five-layer NN with a random noise input layer of size 78 and three internal layers with 128, 128, and 256 nodes respectively, and uses the ReLU activation function for learning the real data distribution. It has an output layer that generates a data sample of size 78, which is comparable to the original data. A generator loss function is computed based on the prediction results provided by the Discriminator and is fed back to the Generator for it to improve in a way to minimize the loss value.

2) Discriminator Configuration: The Discriminator network has an input layer that accepts inputs from the Generator and training dataset during learning phase, and four hidden layers with 128, 128, 64, and 64 nodes respectively, with ReLU activation function and one output layer with sigmoid function to result in an output value between 0 and 1. The output value represents how confident the Discriminator is about the sample being fake or real. Based on this score, the records with a value of less than 0.5 are classified as fake, and are separated from the test data, while the ones classified as real are forwarded to the IDS to get classified further as benign or malign. If the Discriminator correctly classifies a fake sample, the loss function assists the Generator in adjusting its weights to craft more deceptive samples for the Discriminator. If the Discriminator fails to detect a fake sample, the loss function assists the Discriminator in adjusting its weights to improve the detection of fake samples.

3) The Core IDS: This classifier is trained on the training data, and is the IDS model that receives the data that is forwarded by the Discriminator, and decides whether each input is benign or malign.

V. Implementation and Evaluation

A. Preprocessing

The preprocessing stage involves removal of records with NaN and Infinity values, and deletion of the Timestamp column from the dataset. The next step in this stage is to use One-Hot encoding to convert nominal values to numerical data. The CSE-CIC-IDS2018 dataset has a total of 79 features, and after One-Hot encoding, the number of total features becomes 94. Then, Min-Max normalization is applied to all features to scale the data between 0 and 1. This step is essential because the dataset has numeric features whose values might have been derived from various distributions, have varied scales, and are occasionally affected by outliers, which can cause incorrect results by some classifiers. After normalization, 70% of the dataset is separated for training and the remaining 30% for testing. The training set is used to train the ML model. In testing phase, prediction results are obtained by providing the test-set instances as inputs to the model.

To let the model learn from data that has multiple classes, a multi-class classification approach, OneVsRest classification, is followed. OneVsRest is an approach by which the classification algorithms natively designed to perform binary classification can learn from and act on data that has multiple classes. It is achieved by splitting the dataset with multiple classes into multiple datasets with binary classes, and fitting a binary classifier on each of those datasets.

B. Evaluation of Baseline Model in Adversarial Settings

Adversarial samples are generated by implementing the CW attack on the test set, in particular, changeable features in the test data. In simple terms, the way an adversarial algorithm works in an IDS backdrop is by introducing calculated perturbation into a genuine (original) malicious data instance in a way that it gets classified by the target as a benign data/traffic instance while remaining malign. There are no well-defined criteria to determine whether all the generated adversarial traffic instances form meaningful network traffic, however, as rightly explained in [24], some characteristics, as listed below can help understand if perturbations are introduced reasonably:

- the perturbations are added only to changeable features.
- the type of perturbation (binary data, decimal data, etc.) matches the data type of the feature to which it is added.
- semantic relations among the features are preserved.
- the nature of attack an original data instance carries is preserved.
- the perturbation does not disrupt the network-flow information.

The generated adversarial data, along with the original test data is presented to the model for evaluation in adversarial setting. Table 1 presented in Section VI summarizes the results obtained, showing that all the classifiers exhibit a drop in the performance under the influence of the attack.

C. Evaluation of the Defense

The architecture for our defense mechanism uses GAN to identify and separate the adversarial samples, and pass the non-adversarial data to the core IDS classifier, which determines if the data is benign or malign.

During the preprocessing phase, the data is separated into training and testing sets. In training phase, random noise is provided to the Generator to produce fake samples. The Discriminator is trained with the training set and hones its ability to detect adversarial samples by learning the fake data produced by the Generator. The core classifier is also trained with the training set. The GAN, as a whole, becomes more competent as the number of learning/training iterations increase, to not only generate complex fake data but also to identify it correctly. In testing phase, the Discriminator is tested with test data. The test data comprises of the data from test set, and the adversarial samples that are generated by running the CW attack on the test set as explained in Section V.B. The instances that the Discriminator predicts as real are carried forward to the core IDS classifier, separating the instances that are classified as fake.

VI. Evaluation Results and Performance Comparison

Table 1 presents the results provided by the classifier when trained using each of the chosen classification algorithms. They show that there is a clear drop in the performance of baseline classifier when it is tested with adversarial samples. The results from SVM classifier show relatively lesser improvement in the performance than those from the other two classifiers. The results from Table 1 suggest that GAN-based defense enhances the performance as the model can better identify the adversarial data with all of the three classifiers.

TABLE I
Evaluation Results of all the Classifiers

Performance	***Baseline***	***Adversarial***	***Using GAN***
DT Accuracy	0.72	0.49	0.60
DT F1 Score	0.86	0.71	0.67
DT Recall	0.94	0.80	0.65
DT AUC	1	0.99	0.90
RF Accuracy	0.91	0.81	0.82
RF F1 Score	0.94	0.83	0.84
RF Recall	0.91	0.81	0.83
RF AUC	1	0.99	0.90
SVM Accuracy	0.61	0.25	0.36
SVM F1 Score	0.66	0.51	0.43
SVM Recall	0.97	0.88	0.74
SVM AUC	1	0.99	0.90

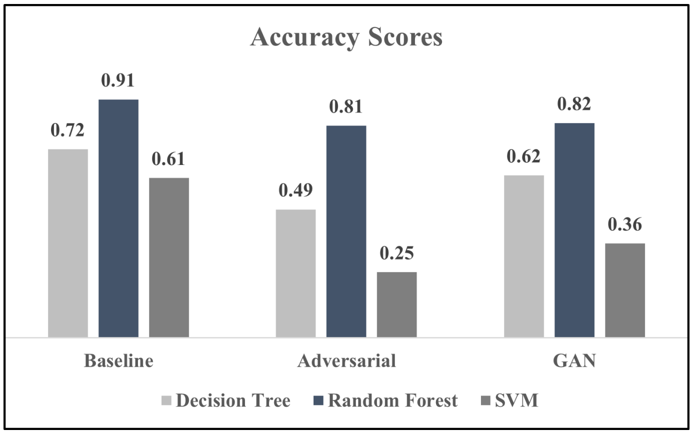

Fig. 3. Comparison of Accuracy scores.

Additionally, we have noted that, on average, 75% of the adversarial data is correctly identified as fake by the Discriminator during the evaluation of the proposed defense. Our aim is to improve the detection abilities further, moving forward. Figure 3 summarizes the improvements in the accuracy scores.

A. Comparison with Related Work

Rui Shu et al. [25] propose a technique called Omni, an ensemble of *unexpected models* to tackle adversarial environments. Their ideology behind employing unexpected models is to keep the distance between their core prediction mechanism and the adversary's target model's mechanism as large as possible. The authors present the experimentation results conducted using five different adversarial white-box evasion attacks on five different cybersecurity datasets. The CW is one of the attacks, and the CSE-CIC-IDS2018 dataset is one among the datasets they have evaluated their approach with. Therefore, we briefly compare our results with the results presented in [25] to analyze our approach further. From the results presented by Rui et al., the baseline accuracy, i.e., under normal settings is 94.48%, and the final accuracy after implementing their defense on the model is 75.23%. The highest baseline accuracy in our approach is 91% and is from the RF classifier, and the corresponding final accuracy (with the proposed defense in place) is 82%. However, since Omni is designed to be agnostic about the type of adversarial evasion attack used, it creates an avenue in our approach to expand its functionalities so as to use it for all kinds of adversarial evasion attacks.

VII. Conclusion

In this paper, we propose a defense mechanism to improve the resistance of ML-based IDSs against the CW attack by using a GAN-based architecture. The dataset used for the work is CSE-CIC-IDS2018, as it reflects a good number of modern network characteristics. The experimentation results show that there is an improvement in the performance of the IDS model with the proposed mechanism. We present the performance results of two scenarios - baseline model alone, and baseline model with GAN. The results indicate that this approach of defense improves the accuracy scores in the adversarial environment created by a powerful white-box attack such as the CW.

We want to take this work forward in multiple directions. One of them is towards improving the detection rate of the proposed GAN-based network in identifying the adversarial data while also investigating how an imbalanced dataset like CSE-CIC-IDS2018 can impact the performance in adversarial environments. We want to extend the architecture by introducing feature squeezing methods and thoroughly evaluate the performance. Furthermore, we plan on expanding the defensive mechanism to tackle black-box attack techniques as they pose a bigger challenge in real-world applications.

References

[1] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, “A deep learning approach for network intrusion detection system,” *Eai Endorsed Transactions on Security and Safety*, vol. 3, no. 9, p. e2, 2016.

[2] M. Usama, M. Asim, S. Latif, J. Qadir *et al.*, “Generative adversarial networks for launching and thwarting adversarial attacks on network intrusion detection systems,” in *2019 15th international wireless communications & mobile computing conference (IWCMC)*. IEEE, 2019, pp. 78–83.

[3] N. Martins, J. M. Cruz, T. Cruz, and P. H. Abreu, “Adversarial machine learning applied to intrusion and malware scenarios: a systematic review,” *IEEE Access*, vol. 8, pp. 35 403–35 419, 2020.

[4] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, “Intriguing properties of neural networks,” *arXiv preprint arXiv:1312.6199*, 2013.

[5] J. Wang, J. Pan, I. AlQerm, and Y. Liu, “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, 2021, pp. 1–9.

[6] A. Thakkar and R. Lohiya, “A review of the advancement in intrusion detection datasets,” *Procedia Computer Science*, vol. 167, pp. 636–645, 2020.

[7] I. Sharafaldin, A. Gharib, A. H. Lashkari, and A. A. Ghorbani, “Towards a reliable intrusion detection benchmark dataset,” *Software Networking*, vol. 2018, no. 1, pp. 177–200, 2018.

[8] S. Dwibedi, M. Pujari, and W. Sun, “A comparative study on contemporary intrusion detection datasets for machine learning research,” in *2020 IEEE International Conference on Intelligence and Security Informatics (ISI)*. IEEE, 2020, pp. 1–6.

[9] N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” in *2017 ieee symposium on security and privacy (sp)*. IEEE, 2017, pp. 39–57.

[10] X. Yuan, P. He, Q. Zhu, and X. Li, “Adversarial examples: Attacks and defenses for deep learning,” *IEEE transactions on neural networks and learning systems*, vol. 30, no. 9, pp. 2805–2824, 2019.

[11] B. Poudel, “Explaining the carlini & wagner attack algorithm to generate adversarial examples.” [Online]. Available: https://medium.com/@iambibek/explanation-of-the-carlini-wagner-c-w-attack-algorithm-to-generate-adversarial-examples-6c1db8669fa2

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *Advances in neural information processing systems*, vol. 27, 2014.

[13] S. Msika, A. Quintero, and F. Khomh, “Sigma: Strengthening ids with gan and metaheuristics attacks,” *arXiv preprint arXiv:1912.09303*, 2019.

[14] S. A. Barnett, “Convergence problems with generative adversarial networks (gans),” *arXiv preprint arXiv:1806.11382*, 2018.

[15] M. Wiatrak, S. V. Albrecht, and A. Nystrom, “Stabilizing generative adversarial networks: A survey,” *arXiv preprint arXiv:1910.00927*, 2019.

[16] Y.-Y. Song and L. Ying, “Decision tree methods: applications for classification and prediction,” *Shanghai archives of psychiatry*, vol. 27, no. 2, p. 130, 2015.

[17] M. Belgiu and L. Drăguţ, “Random forest in remote sensing: A review of applications and future directions,” *ISPRS journal of photogrammetry and remote sensing*, vol. 114, pp. 24–31, 2016.

[18] S. Suthaharan, “Support vector machine,” in *Machine learning models and algorithms for big data classification*. Springer, 2016, pp. 207–235.

[19] Y. Pacheco and W. Sun, “Adversarial machine learning: A comparative study on contemporary intrusion detection datasets.” in *ICISSP*, 2021, pp. 160–171.

[20] A. Ferdowsi and W. Saad, “Generative adversarial networks for distributed intrusion detection in the internet of things,” in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[21] M. H. Shahriar, N. I. Haque, M. A. Rahman, and M. Alonso, “G-ids: Generative adversarial networks assisted intrusion detection system,” in *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*. IEEE, 2020, pp. 376–385.

[22] K. Hara and K. Shiomoto, “Intrusion detection system using semi-supervised learning with adversarial auto-encoder,” in *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2020, pp. 1–8.

[23] N. Papernot, I. Goodfellow, R. Sheatsley, R. Feinman, P. McDaniel *et al.*, “cleverhans v2. 0.0: an adversarial machine learning library,” *arXiv preprint arXiv:1610.00768*, vol. 10, 2016.

[24] H. A. Alatwi and C. Morisset, “Adversarial machine learning in network intrusion detection domain: A systematic review,” *arXiv preprint arXiv:2112.03315*, 2021.

[25] R. Shu, T. Xia, L. Williams, and T. Menzies, “Omni: Automated ensemble with unexpected models against adversarial evasion attack,” *arXiv preprint arXiv:2011.12720*, 2020.