International Journal of Recent Research and Review, Special Issue 2, 2026

ISSN 2277 – 8322

Generative Adversarial Networks for Anomaly and Malware Detection

Amit Kumar Thakore[1], Neha Gupta[2], Akash Saxena[3], Amar Singh Verma[4]

[1 2 4]Department of Computer Science and Engineering, Dr. K. N. Modi University, India

[3]Department of Computer Science and Engineering, CITM, Jaipur, India

Email:vamarverma09@gmail.com,nehaguptanneelkanth@gmail.com,akash27jaipur@gmail.com, amitthakore891@gmail.com

Abstract— With the improvement of computing hardware and the development of large-scale training schemes, generative adversarial networks (GANs) have become a well-known paradigm of deep learning alongside Convolutional and recurrent Neural Networks. Since the first formulation of GAN in 2014, there has been a growing acceptance of such types of models outside of image synthesis, specifically in security sensitive scenarios such as anomaly detection and malware analysis. The paper contains a systematic survey of the application of GAN-based techniques in detecting anomalous and malicious activity in cyber environments. The paper will discuss the main types of GANs, applicable in the area, explain the underlying architecture and training processes, and give a summary of how they are integrated into anomaly and malware detection pipelines. The datasets and evaluation metrics that are publicly available and commonly used in the surveyed work are listed to emphasize the current trends in experimental works and research. Lastly, potential threats and future opportunities of exploiting GANs to counter existent cybersecurity risks are presented with an emphasis on how they can be used to create more resilient responses to them.

Keywords— Generative Adversarial Networks (GANs), mal-ware detection, cyber setting, deep neural networks.

1. INRODUCTION

GANs form a class of deep learning models designed to learn and sample from complex data distributions through adversarial training between a generator and a discriminator. In their standard form, GANs are widely used to train generative models such as deep Convolutional neural networks (CNNs) that can synthesize realistic data, including images and high-dimensional feature representations. By learning from existing samples, GAN-based

ISSN 2277 – 8322

models are able to produce previously unseen but statistically consistent instances, which makes them suitable for augmenting and balancing datasets in security applications. In cybersecurity, this capability is increasingly exploited to construct synthetic corpora that resemble real-world attack traffic or malware behaviors, thereby supporting the design and evaluation of robust defense mechanisms [1]. Malware or malicious software is a code that has been specifically designed to interfere with the normal functioning of the system, steal unauthorized access, or sensitive data. The shift to large-scale remote work during and after the COVID-19 pandemic has coincided with a substantial growth in cybercrime, including large attack surfaces of malware activity. Reports from the industry reveal that in 2021 and 2022, there were billions of recorded malware attacks, with hundreds of thousands of new samples identified each day. The total number of unique malware programs has exceeded one billion, highlighting the extensive and ongoing nature of this threat. Simultaneously, attackers are increasingly uses machine learning approaches to automate and refine their strategies, resulting in advanced and quickly changing threats that pose challenges to conventional detection systems. In this scenario, GANs present a way to reveal" hidden" malicious activities by creating realistic adversarial or rare samples, which can strengthen learning-based detectors against new attack variations [2][4].

The current survey evaluates a wide range of literature that uses GAN architectures to detect and characterize malware, which is important to enhance network and system security. Malware can be described as a versatile attacker toolset, which can be used to carry out different types of actions, including but not limited to disabling a service, elevating privileges, or stealing confidential information without the user's permission. In this context, GANs have been revealed as a two-sided sword: on the one hand, adversarial trained generators can be used to create malware variants that avoid black-box detectors of intrusion and malware with almost zero detection rates; on the other hand, demonstrated that GANs can be utilized to create challenging synthetic samples and that the robustness of intrusion and malware detectors can be enhanced via adversarial training and data augmentation. This paper has identified both of these facets by reporting how GAN-based models are abused to circumvent classifiers and at the same time has shown how they can be incorporated into defensive functions to improve the detection of obfuscated and zero-day threats.

Based on the previous surveys of models

ISSN 2277 – 8322

of GAN variants, the models based on the fields of their application, such as mobile networks, traffic analysis, Internet of Things settings, physical-layer security, and more general cybersecurity work. Specific focus is put on recurring technical issues like unstable dynamics of training, convergence behavior, and mode col-lapse and remedies suggested in terms of better functioning and regularization schemes. This work consolidates to the analyst for contribution of various GAN families to security-related activities in computer and communication networks by mapping GAN architecture and summarizing findings of benefits and limitations of the use of the particular family.

The overall aim of the review is to provide a concise, practice-based body of knowledge of the application of GANs in cybersecurity, focusing on malware and threat-centered research. For researchers and practitioners, the survey serves as a reference point for selecting suitable GAN variants, identifying appropriate datasets and evaluation strategies, and recognizing open problems where GAN-based techniques may be further exploited to counter emerging cyber threats.

2. LITERATURE REVIEW

A Generative Adversarial Network consists of two neural networks trained jointly in an adversarial setting: a generator and a discriminator. The input is a random noise, and the generator is designed to produce artificial samples that are supposed to look like those that come out of the actual distribution, and the discriminator is a binary classifier, which is supposed to identify real samples and produced via the generator.

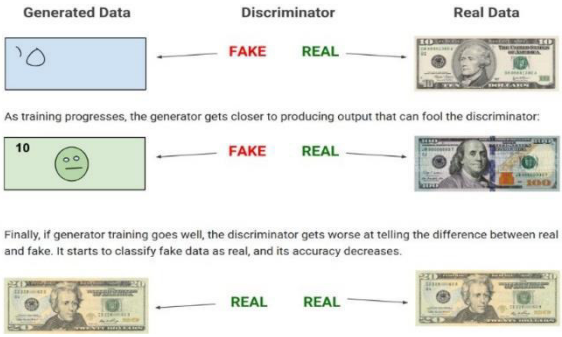


Fig. 1. An image depicting the entire system

Initially, the generator provides the distinct samples which are obviously unrealistic, and their classification as fake by the discriminator is straightforward.

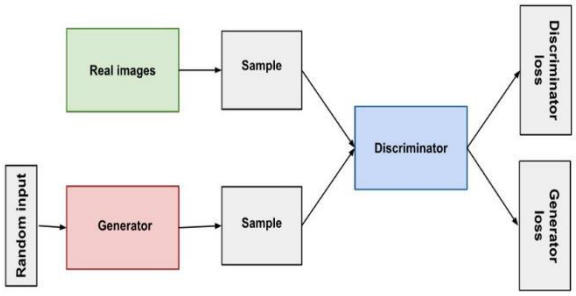


Fig. 2. Generator and Discriminator

The two networks are then updated as training continues: the discriminator modifies its parameters to better its real-fake classification accuracy and the generator modifies its weights according to the feedback of the discriminator so that its out-

ISSN 2277 – 8322

puts are increasingly realistic. This looping process recurs until in a successful training inference the discriminator is unable to draw the difference between generated samples and real data and therefore the generator has learn an approximation of a high fidelity of the target data distribution.

B. Technical Terms and Definitions

GAN is a machine learning model that consists of two separate deep neural networks that are trained in an adversarial environment at the same time. The generator, which is one network, takes random noise as its input and is trained to generate samples, which replicate the properties of real data, and the discriminator; second network measures both authentic and produced samples and labels them as being either real or fake.

The generation process is directed by backpropagation as a minimax problem: when the sample is correctly labeled by the discriminator, the parameters of the discriminator are updated to strengthen this classification, and the generator is informed by a gradient signal that it is necessary to diminish the discriminator so that it is unable to differentiate its products with those of actual data. In contrast, in the case of misclassifications by the discriminator of a sample, future discrimination is enhanced by modification of the parameter. This iterative process continues until the discriminator's accuracy approaches that of random guessing, indicating that the generator has learned to produce samples that are statistically close to the true data distribution.

In this setting, the generator's role is to approximate the underlying data distribution and produce realistic synthetic data, while the discriminator's role is to estimate the probability that a given input originates from the real dataset rather than from the generator. GANs were originally introduced as a type of generative model under the umbrella of unsupervised learning, but subsequent work has demonstrated their applicability to semi-supervised, supervised, and reinforcement learning scenarios through suitable modifications of the loss functions and training protocols.

C. Working principle of GAN

A standard GAN architecture, as illustrated conceptually in Fig. 1, consists of a generator–discriminator pair that forms a two-player zero-sum game, where the generator aims to minimize the discriminator's ability to distinguish real and fake data, and the discriminator aims to maximize this discrimination performance [8]-[10]. In its basic formulation, the model is trained on samples from a target dataset, and the generator learns to map points from a latent noise space to the data space while the

ISSN 2277 – 8322

discriminator learns a decision boundary between real and generated samples.

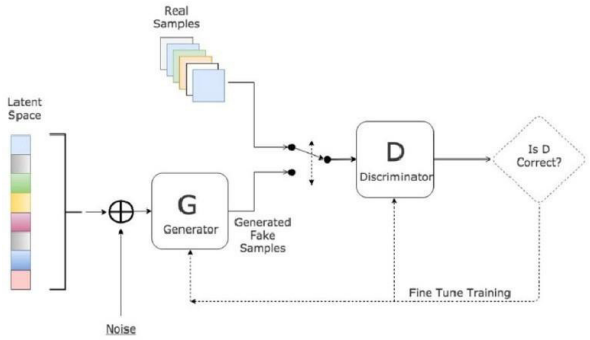


Fig. 3. Generative Adversarial Network

1) Generator: The generator network is responsible for mapping noise vectors to synthetic samples that approximate the real data distribution [12]. It learns to capture the latent structure of the training data so that, over time, its outputs become visually or statistically indistinguishable from genuine examples. During training, the generator is updated to "fool" the discriminator, i.e., to increase the discriminator's probability of labeling generated samples as real.

2) Discriminator: The discriminator is a binary classifier that is trained in a combination of real and synthetic samples of the dataset and the generator respectively [13]. It aims at proper determination of the origin of every input and, therefore, supplying an active learning signal to the generator. The discriminator gets increasingly more refined to differentiate between genuine and fake data as the training goes on, and the internal representation becomes less direct. Generator is adapted to take advantage of the vulnerabilities in this representation.

3) Feedback loop: The connection of the generator and discriminator forms a closed feedback loop which is the focus of the GAN optimization. Derivations of the gradient of the classification loss of the discriminator are also back propagated, not only by the discriminator but by the generator as well, such that both networks can be trained together in a synchronized fashion. Even though it is an effective process by which complex distributions can be learned, this adversarial process is also associated with practical issues, including erratic training dynamics and mode collapse, which have inspired a vast body of theory and practice of better loss functions and regularization methods.

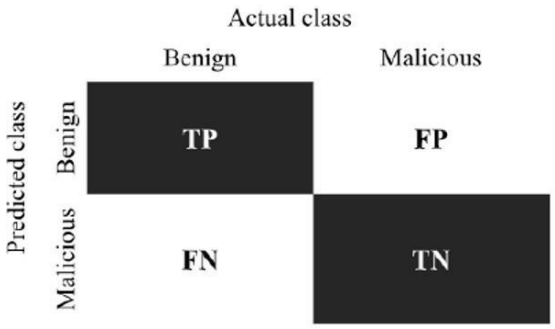


Fig. 4. Machine learning Classifications Performance Metrics

D. Measuring performance

The efficacy of the suggested machine

learning models is measured based on conventional classification measures which are derived according to the confusion matrix i.e. accuracy, precision, recall, F1 and the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) as shown in Fig. 2. These measures provide complementary views of performance and are particularly important in security applications where class imbalance and asymmetric costs of errors are common.

In this context, a true positive (TP) denotes the number of benign samples correctly classified as benign, whereas a true negative (TN) denotes the number of malicious samples correctly identified as malicious. A false positive (FP) corresponds to malicious samples incorrectly labeled as benign, and a false negative (FN) represents benign samples incorrectly labeled as malicious. The overall classification accuracy is computed as the proportion of correctly classified test samples to the total number of test samples, i.e.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Precision quantifies the reliability of positive predictions and is given by

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

Whilst recall (also (as it is also known) as the true positive rate or sensitivity) indicates the proportion of real positive cases that have been successfully identified.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

The F1-score that is the harmonic mean of the recall and the precision values summarizes this trade-off between the recall and the precision,

$$F_1 Score = \frac{2 \times (Precision \times Recall)}{\text{Precision} + \text{Recall}} \quad (4)$$

The metrics of this family is commonly used to measure classification models of security and generative tasks and has been found to provide a high correlation with qualitative measurements in multiple benchmark datasets.

A. Datasets

The studied articles are based on a variety of benchmark datasets to include various dimensions of abnormality and malware-related activity. Among them, Channel State Information (CSI) data of wireless sensing and the Microsoft Malware Classification Challenge corpus are in the center of forming the reported outcomes that are supplemented by network traffic, RF signal, and image-based malware corpus.

1) Wi-Fi RSSI: Wi-Fi Received Signal

Strength Indicator (RSSI) datasets are developed to provide the capacity to localize and identify users according to signal ranges fingerprinting with several access points [27]. They are also applied to smart home automation, internal security surveillance, and occupancy estimation as they may be applied to map RSSI vectors to the locations or occupancy of users within a building.

2) CSI: Channel State Information datasets provide fine-grained measurements of the wireless channel, which are widely used for human sensing tasks such as activity recognition, person identification, and people counting. Recent CSI datasets, for example those collected over 80 MHz IEEE 802.11ac links, record variations induced by human motion across diverse environments, thereby enabling the development and evaluation of machine learning algorithms for behavior recognition and domain-adaptive sensing.

3) Network traffic (KDD Cup '99): The KDD Cup 1999 dataset is a classic intrusion detection benchmark composed of millions of network connection records, each described by 41 features and labeled as normal or as one of several attack types (e.g., DoS, probe, user-to-root, remote-to-local). Despite known limitations and redundancy, it remains a widely used reference for training and evaluating intrusion detection models and for comparative assessment of new techniques.

4) DeepSig RadioML 2016.10A: The RadioML 2016.10A dataset, released by DeepSig, is a synthetic RF dataset generated with GNU Radio and comprising 11 modulation schemes (8 digital and 3 analog) over a range of signal-to-noise ratios. It is primarily used for research on automatic modulation classification and RF signal processing, providing labeled IQ samples for evaluating deep learning-based communication and sensing models.

5) Microsoft Malware Classification Challenge: The Microsoft Malware Classification Challenge dataset is a large-scale benchmark of approximately 20K Windows PE malware samples, provided as raw hexadecimal byte dumps and corresponding disassembly reports [32]. Once uncompressed, the dataset is nearly 0.5 TB in size and covers nine malware families, with each sample labeled by a 20-character identifier and an integer family index; this dual representation supports feature extraction from both byte-level and disassembly-level views and has become a standard testbed for malware family classification and related tasks.

6) Malimg: The Malimg dataset is a collection of 9,339 grayscale images created out of malware binaries of 25 different families. Every binary is converted into a form of an image (e.g., byteplot) to capture

ISSN 2277 – 8322

visual patterns that are typical of individual families and make use of them with image processing and deep learning techniques. Image-based malware detection and classification is also popular in the detection and classification of malware and is frequently utilized with PE or opcode level datasets in experimental research.

TYPES OF GAN

A. Conditional GAN (CGAN)

Conditional Generative Adversarial Networks (CGANs) extend the original GAN framework by conditioning both the generator and discriminator on auxiliary information, such as class labels or attributes. The CGANs are an expansion of the original GAN framework in which the generator and the discriminator receive conditional input (e.g. class labels or attributes) on which the network is trained. The generator of a standard CGAN is re-fed another noise input with an appended condition (e.g. label or feature vector) and the discriminator differentiates between a real and generated input sample of the same condition. It is an architecture enabling the synthesis of sample with desired properties in a controlled way and finds applications in processes such as attribute-conditioned image synthesis, text-to-image synthesis and tabular data synthesis.

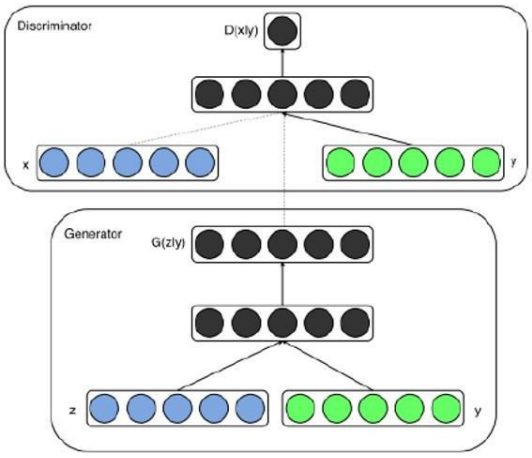


Fig. 5. Architecture of Conditional GAN

B. Wasserstein GAN (WGAN) and WGAN-GP

Wasserstein GANs adopt a variant of Jensen-Shannon divergence-based objective involving a surrogate of the Wasserstein (Earth Mover) distance between the real and the generated distributions that leads to superior training and more accurately predicts the quality of samples. To impose the necessary constraint, the discriminator originally applies weight clipping but it is subsequently replaced by WGAN with Gradient Penalty (WGAN-GP), in which a gradient norm penalty is used in place of hard clipping. These variants are more likely to provide samples of better quality and reduce the issues such as mode collapse compared to vanilla GANs.

C. *RNN-based GANs*

GANs with RNNs apply on the generator or the discriminator or both to process sequential information such as text, time

ISSN 2277 – 8322

series or logs. Architectures like LSTM-GAN that include Long Short-Term Memory units to learn the long-range temporal dependence are found to be effective at generating longer sequences with complex dynamics, whereas simpler RNN-GANs may be useful in the shorter sequences or in situations with low context. These architectures have generative models that generate sequence tokens at a time, and discriminative models that analyses entire sequences and can be trained adversarial at sequence space.

D. Deep Convolutional GAN (DCGAN)

A group of architectural rules regarding stable image generation with CNNs were popularized by Deep Convolutional GANs (DCGANs) introduced by Radford et al. DCGANs replace completely connected with strides convolutes and transposed convolutes, adopt the activation method of batch normalization, and ReLU median generator layer activation with a tan or sigmoid activation method on the final generator layer to put a limit on pixel values. Charismatic Relu irradiation on the discriminator side and Convolutional down sampling is used to boost gradient transfer and studying of depictions. DCGAN has become a foundational architecture for many subsequent image-based GAN variants.

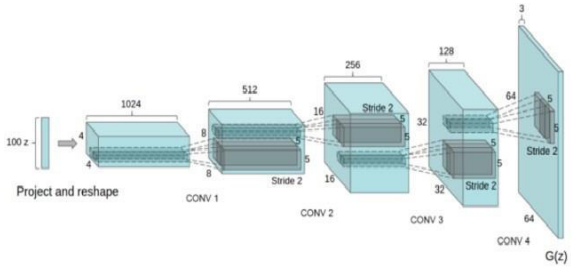
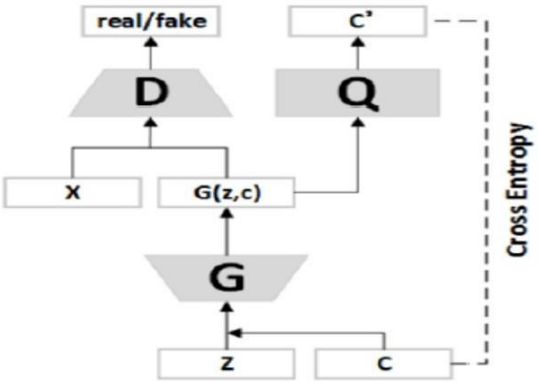


Fig. 6. The Architecture of DCGANs

E. InfoGAN

InfoGAN is an information-theoretic extension of GANs that maximizes the mutual information between a subset of latent variables and the generated observations. The latent vector is decomposed into incompressible noise and "structured" codes, and an auxiliary network is introduced to approximate the posterior over these codes, enabling an additional mutual-information regularization term in the objective [36]. This trained the model to learn disentangled and interpretable latent factors which has been shown to work on datasets like MNIST, CelebA, and SVHN where latent codes differing between samples result in meaningful changes in the digit style, face, or object appearance.



ISSN 2277 – 8322

Fig. 7. The Architecture of InfoGAN

F. *CycleGAN*

CycleGAN [38] is an image-to-image translation framework that learns mappings between two visual domains using unpaired training data. It employs two generators and two discriminators, together with a cycle-consistency loss that enforces that translating an image from one domain to the other and back reconstructs the original input. This design enables translation tasks such as style transfer, domain adaptation, and object appearance modification without requiring aligned image pairs, extending earlier paired approaches such as Pix2Pix [39].

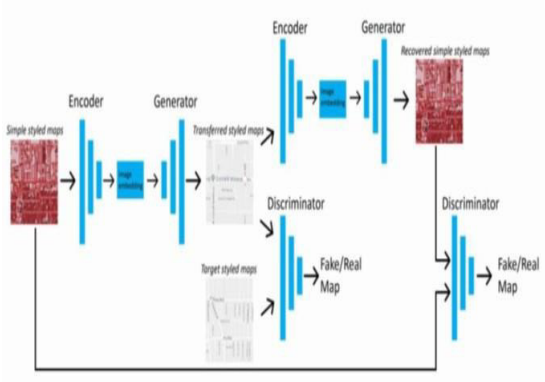


Fig. 8. The Architecture of CycleGAN

G. Auxiliary Classifier GAN (AC-GAN)

Auxiliary Classifier GAN (AC-GAN) augments the discriminator with an additional classification head that predicts class labels for both real and generated samples. The generator is conditioned on class information and is trained to produce samples that are both realistic and class-discriminative, while the discriminator jointly optimizes an adversarial loss and a supervised classification loss [17][41]. Compared with CGAN, which conditions implicitly through concatenation, AC-GAN explicitly structures the latent space via the auxiliary classifier, often improving class-conditional image quality and label controllability.

H. Temporal GAN (TGAN)

Temporal Generative Adversarial Networks (TGANs) are enthusiastic about video and other time-dependent data by explicitly modeling the time dimension [44]. A generic TGAN uses a temporal generator to obtain a sequence of latent codes representing frames in a video and an image generator to enable each latent code to correspond to a frame to enable the model to cover both dynamics over time and structure in space. Numerous TGAN models use the Wasserstein-type objectives and end-to-end training frameworks to stabilize training and have been shown to be able to generate various and coherent video sequences on unlabeled input.

I. *Least Squares GAN (LSGAN)*

Least Squares GANs modify the discriminator's loss from a sigmoid cross-entropy objective to a least-squares loss, which helps mitigate vanishing gradients and

ISSN 2277 – 8322

encourages generated samples to move closer to the decision boundary in a more controlled manner [44]. This modification regularly results in smoother gradients of the generator, which gives rise to more superior images and more consistent training conduct. LSGANs have been successfully applied in unsupervised and conditional generation tasks as an alternative to the original GAN loss.

I. Evolutionary GAN (E-GAN)

Evolutionary Generative Adversarial Networks (E-GANs) is an extension of GANs that uses the idea of evolutionary computation in training GANs by keeping a pool of generators 'G' and evolves them with time. Every iteration involves

$$L_G = L\ G(z_G), \text{ for } \theta_G$$

$$k_{t+1} = k_t + \lambda_k\ \gamma L(x) - L\ G(z_G),$$

for each training step t is updated at each step to maintain a target ratio between real and generated reconstruction losses, promoting stable training and high-quality image synthesis.

J. Progressive Growing GAN (ProGAN)

Progressive Growing of GANs (ProGAN), proposed by Karras et al., trains GANs by gradually increasing the resolution of generated images and discriminated inputs. Training begins at a low spatial resolution, where the generator and discriminator learn core structure, and new layers are progressively added to both networks to handle higher resolutions, with smooth transition phases to avoid instability. Combined with improved loss functions such as WGAN-GP and LSGAN, this strategy enables the generation of high-resolution, photorealistic images and has influenced many subsequent large-scale generative models.

K. MSG-GAN

Multi-Scale Gradient GAN (MSG-GAN) addresses gradient flow and stability issues in high-resolution generation by allowing gradients from the discriminator to be propagated to the generator at multiple image scales. Instead of only feeding the final high-resolution output to the discriminator, intermediate generator feature maps at different resolutions are also connected to corresponding discriminator branches. It is a multi-scale feedback which enhances signal propagation, diminishes the effects of noisy gradient, and produces sharper, more consistent high-resolution images as an alternative to entirely progressive growing schemes.

3. TYPES OF GAN-BASED DETECTIONS

A. Anomaly Detection

ISSN 2277 – 8322

The use of GAN as an anomaly-detecting tool has been gaining popularity as the method of identifying attacks grows more intricate and difficult to describe using traditional mod-els. Earlier experiments by Navidan et al. [17] give a general taxonomy of types of anomaly and detection paradigms, which encourages the application of the more sophisticated learning-based techniques in this field. Subsequently, Wang et al. [46] proposed AnoGAN, which is a DCGAN-based architecture

[13] that is trained on normal data in a two-gan setup, and detects anomalies when presented with test data using both reconstruction error metrics and latent-space consistency metrics.

In order to mitigate scalability and inversion problems of AnoGAN, Odena et al. [37] introduced Efficient GAN-Based Anomaly Detection (EGBAD), a framework that in-corporates the BiGAN [42] framework to learn an explicit encoder in the input space to latent space through adversarial training, enabling the framework to save training time and score anomalies. Another model was introduced by Donahue et al. [39] GANomaly, which combines an encoder-decoder-encoder generator with a discriminator and a set of loss terms, demonstrates better convergence behavior on a variety of image benchmarks, including a higher count of discriminative latent representations of normal data and improved performance.

B. Cyber Intrusion and Malware Detection

GANs have also been used in cybersecurity to stress-test and in intrusion detection system (IDS) strength. Some studies reveal that adversarial traffic generated by GANs can take advantage of vulnerabilities in ML-based IDSs, which implies their vulnerability to engineered instances and implies that this type of defense should be strong [16]. As a case study, (GAN-based) attacks have been trained to learn manipulations of traffic characteristics to confound the IDS models into treating malicious flows as benign, and follow-up research proposes using GAN-assisted data augmentation and hybrid models to improve the robustness of the benchmarks, such as UNSW-NB15 and CIC-IDS-2017.

GANs have been used in the malware field in various plat-forms, including windows and Android, to generate, transform, and recognize malicious binaries. Image based techniques represent malware binaries as grayscale or RGB images and train GANs or DCGAN-like networks to aid in detections and family classification like PlausMal-GAN and others that solve zero-day and class-imbalance problems. Other work, such as TDCGAN-style models,

ISSN 2277 – 8322

explores transfer learning and adversarial generation to detect previously unseen malware variants, illustrating the adaptability of GANs to different operating systems and representation schemes (e.g. PE files, opcode sequences, behavior logs). These contributions collectively show that GANs can be used to perturb traffic, augment datasets, and enhance intrusion and malware detectors.

C. *Security Attacks*

Beyond detection, GANs are increasingly utilized for testing and hardening systems. One prominent application is the generation of realistic but synthetic attack payloads that support vulnerability assessment and automated penetration testing. Chowdhary et al. proposed an autonomous web application penetration testing framework that employs semantic tokenization to extract salient features from XSS and SQL injection payloads and uses a conditional sequence GAN to produce new attack strings capable of evading web application firewalls at scale.

By learning the distribution of existing exploits and generating diverse variants, such GAN-based frameworks can systematically probe web applications and services for weaknesses associated with advanced persistent threats (APTs), input-validation flaws, and other classes of vulnerabilities. These attempts show the dual-use of GANs in security: as they are the means of building complex offensive scenarios and as they are the facilitators of the more stringent and automated testing of defensive mechanisms.

4. CONCLUSIONS

This study provides a comprehensive review of Generative Adversarial Networks in malware analysis and computer security in general, and the focus on the tasks of anomaly and threat detection. The work provides a unified gateway to improving the comprehension of how GAN-based approaches can be applied in cyber defense by bundling the basics of GANs, listing the variants of these algorithms as applied in security tasks, and reevaluating those evaluation metrics that have become widely accepted in practice.

The analysis indicates the existence of gaps in existing research, such as training instability, lack of standardization of benchmarks, and dual use natures of GANs and emphasizes the significance of hyper parameter tuning as the way to enhance the detection accuracy and robustness. Also, the paper presents the bright future research opportunities, including use of GANs in more complex neural cryptographic, more detailed attack and traffic representations, and hybrids between GANs and other deep learning and graph-based models to overcome the current cybersecurity

mechanisms. Having presented a systematic review of GAN applications in malware detection, intrusion detection, and anomaly detection, the survey is aimed at helping researchers and practitioners to choose the right GAN type and the dataset, as well as design decisions. By so doing, it will enable more orderly evolution of GAN enhanced security solutions, and will spur more progress in resilient network and system protection.

REFERENCES

[1] Bayer et al. "Dynamic Analysis of Malicious Code" vol. 2, pp. 66-67, 2006; Doi: 10.1007/s11416-006-0012-2.

[2] A. Petrosyan, "Annual number of malware attacks worldwide from 2015 to 2022" 2023. Online, Accessed 2 2 2024.

[3] N. J. Palatty "30+ Malware Statistics You Need To Know In 2024" Astra, 2023. Online, Accessed 2 2 2024.

[4] J. Brownlee, "How to develop an auxiliary classifier GAN (AC-GAN) from scratch with Keras" 2021. Online, Accessed 2 2 2024.

[5] Dunmore et al. "Generative Adversarial Networks for Malware Detection: a Survey" ArXiv, vol. abs/2302.08558, 2023; Doi: 10.48550/arXiv.2302.08558.

[6] S. Gihon, "Ransomware Trends Q4 2023 Report" 2024. Online, Ac-cessed 2 2 2024.

[7] W. Hu and Y. Tan, "Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN" ArXiv, vol. abs/1702.05983, 2017.

[8] Salimans et al. "Improved Techniques for Training GANs" ArXiv, vol. abs/1606.03498, 2016.

[9] Isola et al. "Image-to-Image Translation with Conditional Adver-sarial Networks" in 2017 IEEE Conference on Computer Vi-sion and Pattern Recognition (CVPR), 2017, pp. 5967-5976; Doi: 10.1109/CVPR.2017.632.

[10] J. Ho and S. Ermon, "Generative Adversarial Imitation Learning" in Neural Information Processing Systems 2016.

[11] A. Gharakhanian, "Generative Adversarial Networks – Hot Topic in Machine Learning" 2017. Online, Accessed 2 2 2024.

[12] He et al. "Analysis of Image Generation by different Generator in GANs" Journal of Physics: Conference Series, vol. 1903, 2021.

[13] Radford et al. "Unsupervised Representation Learning with Deep Convo-lutional Generative Adversarial

Networks" CoRR, vol. abs/1511.06434, 2015.

[14] Goodfellow et al. "Generative Adversarial Nets" in Neural Information Processing Systems, 2014.

[15] M. Arjovsky and L. Bottou, "Towards Principled Methods for Training Generative Adversarial Networks" ArXiv, vol. abs/1701.04862, 2017.

[16] Cai et al. "Generative Adversarial Networks" ACM Computing Surveys (CSUR), vol. 54, pp. 1-38, 2021.

[17] Navidan et al. "Generative Adversarial Networks (GANs) in Net-working: A Comprehensive Survey & Evaluation" ArXiv, vol. abs/2105.04184, 2021.

[18] Won et al. "PlausMal-GAN: Plausible Malware Training Based on Generative Adversarial Networks for Analogous Zero-Day Malware Detection" IEEE Transactions on Emerging Topics in Computing, vol. 11, pp. 82-94, 2023; Doi: 10.1109/TETC.2022.3170544.

[19] Dutta et al. "Generative Adversarial Networks in Security: A Survey" 2020. 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), pp. 0399-0405, 2020.

[20] Prabakaran et al. "An enhanced deep learning-based phishing detection mechanism to effectively identify malicious URLs using variational autoencoders" IET Information Security, vol. 17, no. 3, pp. 315-551, 2023; Doi: 10.1049/ise2.12106.

[21] Salimans et al. "Improved Techniques for Training GANs" ArXiv, vol. abs/1606.03498, 2016.

[22] S. T. Barratt and S. Rishi, "A Note on the Inception Score" ArXiv, vol. abs/1801.01973, 2018.

[23] A. Borji, "Pros and Cons of GAN Evaluation Measures" ArXiv, vol. abs/1802.03446, 2018.

[24] Dunmore et al. "A Comprehensive Survey of Generative Adversarial Networks (GANs) in Cybersecurity Intrusion Detection" IEEE Access, vol. 11, pp. 76071-76094, 2023; Doi: 10.1109/ACCESS.2023.3296707.

[25] Che et al. "Mode Regularized Generative Adversarial Networks" ArXiv, vol. abs/1612.02136, 2017.

[26] Szegedy et al. "Rethinking the Inception Architecture for Computer Vi-sion" in 2016 IEEE Conference on

Computer Vision and Pattern Recog-nition (CVPR), 2016, pp. 2818-2826; Doi: 10.1109/CVPR.2016.308.

[27] Rohra et al.“User Localization in an Indoor Environment Using Fuzzy Hybrid of Particle Swarm Optimization & Gravitational Search Al-gorithm with Neural Networks” in International Conference on Soft Computing for Problem Solving, 2016.

[28] Meneghello et al. “A CSI Dataset for Wireless Human Sensing on 80 MHz Wi-Fi Channels” IEEE Communications Magazine, vol. 61, pp. 146-152, 2023.

[29] Yousefi et al. “A Survey on Behavior Recognition Using WiFi Channel State Information” IEEE Communications Magazine, vol. 55, pp. 98-104, 2017.

[30] Ronen et al. “Microsoft Malware Classification Challenge” ArXiv, vol. abs/1802.10135, 2018.

[31] Nataraj et al. “Malware images: visualization and automatic classifica-tion” in Visualization for Computer Security, 2011.

[32] M. Mirza and S. Osindero “Conditional Generative Adversarial Nets” ArXiv, vol. abs/1411.1784, 2014.

[33] Chen et al. “InfoGAN: Interpretable Representation Learning by Infor-mation Maximizing Generative Adversarial Nets” in Neural Information Processing Systems, 2016.

[34] Li et al. “SCGAN: Disentangled Representation Learning by Adding Similarity Constraint on Generative Adversarial Nets” IEEE Access, vol. 7, pp. 147928-147938, 2019.

[35] Zhu et al. “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks” IEEE International Conference on Computer Vision (ICCV), pp. 2242-2251, 2017.

[36] Isola et al. “Image-to-Image Translation with Conditional Ad-versarial Networks” 2017 IEEE Conference on Computer Vi-sion and Pattern Recognition (CVPR), pp. 5967-5976, 2016; Doi: 10.1109/CVPR.2017.632.

[37] Odena et al. “Conditional Image Synthesis with Auxiliary Classifier GANs” in International Conference on Machine Learning, 2016.

[38] R. Nagaraju and M. Stamp “Auxiliary-Classifier GAN for Malware Analysis” ArXiv, vol. abs/2107.01620, 2021.

[39] Donahue et al. “Adversarial Feature Learning” ArXiv, vol. abs/1605.09782,

ISSN 2277 – 8322

2016.

[40] Xu et al. “Improved bidirectional GAN-based approach for network intrusion detection using one-class classifier” MDPI (Basel, Switzerland) 2022; Doi: 10.3390/computers 11060085.

[41] Saito et al. “Temporal Generative Adversarial Nets with Singular Value Clipping” in 2017. IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2849-2858; Doi: 10.1109/ICCV.2017.308.

[42] Munoz et al. “Multi-Variate Temporal GAN for Large Scale Video Generation” ArXiv, vol. abs/2004.01823, 2020.

[43] Mao et al. “Least Squares Generative Adversarial Networks” in 2017. IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2813-2821; Doi: 10.1109/ICCV.2017.304.

[44] Arjovsky et al. ”Wasserstein GAN,” ArXiv, vol. abs/1701.07875.

[45] Gulrajani et al. “Improved Training of Wasserstein GANs” in Neural Information Processing Systems, 2017.

[46] Wang et al. ”Evolutionary Generative Adversarial Networks” IEEE Transactions on Evolutionary Computation, vol. 23, pp. 921-934, 2019; Doi:10.1109/TEVC.2019.2895748

[47] Berthelot, et al. “BEGAN: Boundary Equilibrium Generative Adversar-ial Networks” ArXiv, vol. abs/1703. 10717, 2017.