IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 27 June 2023, accepted 14 July 2023, date of publication 19 July 2023, date of current version 27 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3296707

SURVEY

A Comprehensive Survey of Generative Adversarial Networks (GANs) in Cybersecurity Intrusion Detection

AERYN DUNMORE[1], **(Graduate Student Member, IEEE), JULIAN JANG-JACCARD**[1], **FARIZA SABRINA**[2], **(Member, IEEE), AND JIN KWAK**[3]

[1] Cybersecurity Laboratory, Department of Computer Science, Massey University, Auckland 0653, New Zealand
[2] School of Engineering and Technology, Central Queensland University, Sydney, NSW 4701, Australia
[3] Department of Cyber Security, Ajou University, Suwon, Gyeonggi-do 206 KR South Korea

Corresponding author: Aeryn Dunmore (a.dunmore@massey.ac.nz)

This work was supported by the Ministry of Business, Innovation, and Employment (MBIE) from the New Zealand Government under Grant MAUX1912.

ABSTRACT Generative Adversarial Networks (GANs) have seen significant interest since their introduction in 2014. While originally focused primarily on image-based tasks, their capacity for generating new, synthetic data has brought them into many different fields of Machine Learning research. Their use in cybersecurity has grown swiftly, especially in tasks which require training on unbalanced datasets of attack classes. In this paper we examine the use of GANs in Intrusion Detection Systems (IDS) and how they are currently being employed in this area of research. GANs are currently in use for the creation of adversarial examples, editing the semantic information of data, creating polymorphic samples of malware, augmenting data for rare classes, and much more. We have endeavored to create a paper that may act as a primer for cybersecurity specialists and machine learning researchers alike. This paper details what GANs are and how they work, the current types of GAN in use in the area, datasets used in this research, metrics for evaluation, current areas of use in intrusion detection, and when and how they are best used.

INDEX TERMS Generative adversarial networks (GAN), machine learning, research survey, attack modeling, threat detection, intrusion detection systems, data augmentation, zero-day attacks, adversarial examples.

I. INTRODUCTION

The Generative Adversarial Model, or GAN, is a method proposed by Goodfellow et al. [1] in 2014 as a new alternative to Variational Autoencoders (VAE) [2] for generating large amounts of synthesized but realistic data. The power behind the GAN model and the research it has spurred on, is the ability to augment and even create datasets, a talent greatly in demand due to the ever-rising tide of machine learning-driven technology. Training machine learning models requires a substantial dataset, necessitating human collated and labeled datasets which are expensive in both cost and in time. While Google has used programs like reCAPTCHA[1] to create large labeled datasets for computer vision [4], most do not have a similar opportunity to leverage the average citizen for creating datasets. As a result of this lack of data, models like GAN or VAE are looked at to help train new machine learning systems. In security, GAN models can be very useful at generating samples of malicious code, traffic, or behavior. As a result, these models are being employed with great success in research towards new or improved Intrusion Detection Systems. Our aim with this paper is to survey the current state of the art in utilizing GAN models for Intrusion Detection Systems challenges and research. We endeavor to present both breadth of topics and depth of knowledge,

The associate editor coordinating the review of this manuscript and approving it for publication was Jemal H. Abawajy.

[1] Alphabet, Google's parent company, confirms the use of your answers for purposes other than verification of your status as a human [3].

that we might offer a contribution which is of use to both the experienced machine learning researcher as an update on current research and methods, and also as a primer to those entering into GAN research for cybersecurity. We have also done our utmost to cover these topics from the point of view of both cybersecurity and machine learning researchers. We have offered a comprehensive review of not just the current research, but the datasets used for testing, the methods and designs of the GAN models used in experimentation, and the metrics used to evaluate both.

The remainder of our paper is structured as follows: Section II introduces the reader to the basic structure of the original Generative Adversarial Networks proposed in [1], as well as defining Intrusion Detection Systems and malicious operators for the purpose of our survey. Section III will explain the other survey papers in the area, and show how we have filled a knowledge gap in the specificity of subject and depth of knowledge, while Section IV details the metrics by which researchers evaluate their schemes. Section V explains the datasets on which the models have been trained and tested. Section VI goes into detail about the different variants and models of GANs that have been proposed in the papers we have surveyed. Section VII investigates the uses that are current hot topics in research, and then Section VIII discusses the research applications in detail. Finally, Section IX goes into the potential avenues for future research, and Section X concludes our survey.

II. EXPLANATIONS, TERMS, AND THE GAN MODEL

A. WHAT IS GAN?

The basic Generative Adversarial Network model, as proposed by Goodfellow et al. [1], is a two-network, two-player game, with a zero-sum target based in Game Theory. The Generator, which is trained on a dataset of real samples, tries to generate convincing samples which can fool the Discriminator, such that the Discriminator believes the samples are genuine. They are considered *semi-supervised learning*, and the weights are adjusted through back-propagation. The game is over when the Discriminator can only tell the real samples from the generated ones with an accuracy of 50%, effectively making a binary guess or a coin toss. The generation of new samples that are all-but-real makes GAN models very desirable. In order to be able to train Intrusion Detection Systems, Antivirus, and other defensive technologies, to detect when a communication, file, or action, is malicious, large sets of classes and data types with many samples are needed. It is simple to see how this can give GAN models a significant place in research in security going forwards.

1) GENERATORS

The Generator Network in the model is the more complicated of the two. It starts, in training a "vanilla" GAN (the original, Goodfellow model), with a random seed, sometimes referred to as a noise sample, and then the Generator begins generating samples immediately. These early attempts are very unsuccessful, as they are primarily random noise, but as more and more feedback propagates backwards from the Discriminator, the Generator slowly improves the quality of its samples, bringing them closer to the genuine samples the training set contains. The Generator is also the part of the model that is generally kept after convergence is achieved or the full number of training epochs has run [5]. Once training is complete and the Generator is capable of synthesizing samples that are all but genuine, it is ready to be used for the purpose for which it was built. In some cases, the Generator fails to win against the Discriminator, which instead becomes a highly effective classifier. Some research scenarios utilize the Discriminator for this purpose, rather than discarding it.

2) DISCRIMINATORS

As discussed above, the Discriminator of the model is not usually kept after the Generator has been successfully trained [6]. The essence of the Discriminator is to look at samples provided by the Generator, both genuine and synthesized, and successfully categorize them. As the Generator gets the feedback and slowly alters its weights for more accurate sample generation, the Discriminator is supposed to slowly become less and less effective, until it is little more than a computerized coin toss. In some cases, the opposite occurs, with the Generator unable to model the provided samples accurately. In the original proposed model however, the Generator wins the game. At this point, the Discriminator is no longer necessary, as it has fulfilled the purpose for which it was built - training the Generator to synthesize exceptionally realistic samples. As mentioned earlier, sometimes the Discriminator does win the game and, depending on the type of GAN model, this can result in a useful and accurate classifier. In some models, the Generator is creating labeled samples of different classes, meaning the Discriminator is carefully trained to know what each class of samples should look like.

3) CONVERGENCE

In Equation 1, we have provided the minmax game that sits at the heart of the GAN model. The $p_z(z)$ input contains the z variable, the seed data for the Generator, while the p function plots a noise distribution. $V(D, G)$ provides the value function in which G is the Generator, with the value function of $G(z; \theta_g)$ and D is the discriminator. The result of the Discriminator's function $D(x; \theta_D)$ is the probability value (a single scalar value), which suggests whether the input, x, came from the training set or from the Generator. Because both networks are being trained concurrently, the goal is to minimize the $log(1 - G(z))$ for training the Generator, while also minimizing $log(D(x))$ for the Discriminator [7]. Once the probability value - the output scalar from the value function of the Discriminator - flattens into 0.5, the game is over and convergence has been achieved.

$$\underset{G}{min}\,\underset{D}{max}\, V(D, G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] \qquad (1)$$

$$+ \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \qquad (2)$$

B. INTRUSION DETECTION SYSTEMS

The central tenants of cybersecurity are Confidentiality, Integrity and Availability (CIA). An Intrusion Detection System (IDS) is, at heart, a method for ensuring the strongest possible version of the CIA requirements for its host or user [8]. IDS models are not new - in 1988, Smaha [9] proposed an IDS called Haystack. Knowing if there has been an attempt on your device, successful or not, is a particularly important part of keeping yourself safe, be you a person on the street with a smartphone, or a giant corporation with its own server farm. IDS models are meant to enforce the CIA protocols that are the core of cybersecurity training. An IDS model is built to detect unauthorized user behavior, and/or to detect behavior from authorized users that falls outside the purview of their authorization [10]. In simplest terms, an IDS should be able to tell if traffic is malicious or legitimate [11]. An intrusion, for the purpose of this paper and research, is an effort or instance of attempting to circumvent or cause a failure in CIA [8]. Modern IDS models are either Network-based (in that they monitor the packets exchanged) or Host-based (in that they monitor logged behavior on a device) [12]. An extension of IDS is the *Intrusion Prevention System*, or IPS, which takes the behavior of an IDS one step further as an attempt to shield the host from unauthorized access attempts. Given this, it is not surprising that machine learning for IDS models has taken off with such gusto. Of course, the enduring problem with machine learning models is their training phase and the expanse of data required to create the necessary training and testing datasets. According to Thakkar and Lohiya [11], there are a number of dangers in network traffic to consider at this point in the evolution of internet usage. These include:

- Attempts to obtain personal and private data
- Ransomware
- Adversarial AI
- IoT-focused attacks

Machine learning techniques for IDS models belong to the category of *Anomaly-based Detection*. Traditional methods of IDS systems also include Signature and Stateful Protocol Analysis detection methods [8]. IDS models can also be divided along the type of classification provided - a binary classifier will assign a class of either attack or benign, while a multiclass classifier may offer more detailed classification, such as the type of attack. Rare classes or types of attacks have few samples, while benevolent or normal traffic is plentiful. Alongside this is the problem of testing your IDS against an enemy actor. These are just some of the ways researchers are using Generative Adversarial Networks for the building of IDS models.

III. RELATED WORK

This section attempts to create a general overview of the surveys on Generative Adversarial Networks in cybersecurity and with regards to intrusion detection. We briefly detail the paper and the topics discussed, as well as where we feel our survey fits within the current research landscape.

Arora and Shantanu [13] reviewed uses for Generative Adversarial Networks in the cybersecurity domain. This survey does delve into the types of GAN models, explaining the different types of models and giving graphical representation of these models, as seen in Figure 3. The paper also places a lot of emphasis on a case study of anomaly detection and generation using the KDD-NSL dataset. While it offers a good overview of some of the different GAN models, it lacks in both depth and breadth of applications, with a specific focus on network intrusion, and some exploration on steganography and password guessing. Though this paper is timely and important, we do not believe it negates the need for our paper, as the authors survey only a small number of GAN models - vanilla GAN, DCGAN, BiGAN, and CycleGANs. The paper looks at the types of datasets in use, and more specifically the individual domains of cybersecurity in which GAN models can be used. Because that paper is so compact, it does not have the opportunity to go into depth in the way we do in this survey.

Dutta et al. [14] did an extensive survey paper that explores many different types of algorithms using the GAN model, for security purposes. It shows both defensive and offensive algorithms, to balance the paper with the ways GANs can be applied in the security domain.

In [14], the authors are careful to extend a wide range of spaces in which GANs could be used to improve the security of sensitive information. Amongst other areas, healthcare and banks. The authors also discuss ethics and possible misuse of technology. Overall, this paper does raise some very interesting studies, and the survey covers a wide range of topics. However, it is quite a short survey, and one thing noticeably absent in most sections is any type of metric for the study in question. We found this unusual for a survey paper which discusses the significance and successes of the use of GAN in the cybersecurity domain.

Cai et al. [15] have created a highly detailed and in-depth survey of the elements of security and privacy wherein GAN can be applied. This paper is very insistent on showing *both* sides of GAN research. Cases wherein the generator is an attacker against the defending classifier (such as [16], [17], [18], [19]), as well as those wherein the generator is defending itself against the attacking discriminator (such as [20], [21], [22], [23], [24]) are examined. The latter include GAN models such as Generative Adversarial Privacy (GAP), Privacy Preserving Adversarial Networks (PPANs), Compressive Adversarial Privacy (CAP), and Reconstructive Adversarial Network. A point of interest in the paper is the section surveying "model" privacy. "A model's privacy breaches if an adversary can use the model's output to infer the private attributes used to train the model." (pp. 132:13, [15]) We have found that other surveys do not include this as standard in the sections of their papers on security. The importance of model privacy seems to normally be an overlooked one. This paper takes the time to look at it, with a definition based on [25]. While it is an intriguing area of research for security and privacy, we do not believe that it is necessary to go into the same level of detail in our own survey.

We believe our survey has found a space within these existing surveys to fill gaps with regards to how Generative Adversarial Networks are used in building an effective IDS model. We have done this while focusing on creating an overview that will be of use to researchers in machine learning and Intrusion Detection, new and experienced. The survey papers' topics and specific GAN models are summarized in Table 1.

IV. MEASURING PERFORMANCE

The used performance metrics for evaluating Machine Learning are a very select and oft-repeated set. Here, we try to ensure that our reader is as familiar with these metrics.

1) TRUE POSITIVE

A true positive (TP) occurs when the model correctly identifies a benign sample as benign.

2) FALSE POSITIVE

A false positive (FP) occurs when a model classifies a malicious sample as benign.

3) TRUE NEGATIVE

A true negative (TN) occurs when the model classifies a malicious sample as malicious.

4) FALSE NEGATIVE

A false negative (FN) occurs when the model incorrectly classifies a benign sample as malicious.

Using TP, FP, TN, FN metrics is only a small part of measuring the performance of the machine.We will now introduce some methods of measuring performance that go slightly deeper. Some papers do not press much farther than the above, however the best methods for determining a particular model's success may be different to those of another.

a: ACCURACY

The accuracy of a model is the overall mean of the predictions made by the model, both correct and incorrect. It measures the total *correct* predictions against the total predictions both correct and incorrect.

$$acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

b: PRECISION

The Precision or Positive Predicted Value (PPV) is the measurement of all the true positive predictions, against all the predictions of a positive class, both TP and FP. In this way, it evaluate the overall ways in which the model successfully or unsuccessfully classes the positive values.

$$P = \frac{TP}{TP + FP} \tag{4}$$

c: RECALL

The recall, also called the True Positive Ratio, or the sensitivity, of the model, is classified as the number of true positives predicted by the model, over the number of true predictions overall, both positive and negative.

$$R = \frac{TP}{TP + TN} \tag{5}$$

d: HARMONIC MEAN

The Harmonic Mean, also known as the F1-Score, is the method by which the performance of a model is measured with regards to its minority class. This is especially important in cases such as those involving classification and neural networks. The ability to accurately classify the class which occurs the least in the training set, that is the rarest of the samples, is both extremely important and extremely difficult. This is calculated as the trade-off between Precision (P) and Recall (R), as shown below.

$$F_1 = 2\left(\frac{PR}{P + R}\right) \tag{6}$$

e: THE INCEPTION SCORE

The Inception Score is one of the less common methods of measuring the performance of a model. It determines the distribution of the model's predictions and classifications as the distribution of probabilities over two sets of distributions, Ω_X and Ω_Y [26]. When g is classed as the Generator, and we label d as the Discriminator, we have the distribution of the generator as p_g, and the Discriminator function can be determined as $p_Y : \Omega_x \rightarrow M(\Omega_Y)$. We classify each image as some x, and each label as some y. We have the set of all possible distributions $M(\Omega_Y)$, over the set Ω_Y. We can then go on to say that writing $p_d(y|x)$ is writing the function that gives the probability that a given x has the label y. The Inception Score was originally developed for computer vision tasks - the equation offers as an output a value in the range $[1, 1000]$, with the higher values meaning a higher level of quality or detail in an image [27]. It came to use in CNN models in [28].

f: MODE SCORE

A modified version of the Inception Score, the Mode Score [29] is designed to ignore the distribution of the original set of probabilities. Introduced in [30], the Mode Score was designed to deal with "missing modes", or areas in which the generator was undertaking very little sampling and where the discriminator therefore took precedence. The mode score was also designed in order to offer a way to evaluate sample quality without a human annotator.

g: FRÉCHET INCEPTION DISTANCE

Another evaluative metric designed originally for use in computer vision tasks [28], the FID, as a version of the original Inception Score, is designed as an attempt to combat

TABLE 1. Types of GAN research in related works.

Topics	Arora, et al., 2022 [13]	Dutta et al., 2020 [14]	Cai, Z. et al., 2021 [15]
Intrusion Detection Systems		X	X
Malware		X	X
Adversarial Examples	X		
Reinforcement Learning	X		
Offensive/Attacker Models		X	X
Defensive/Defender Models		X	X
Privacy Preserving Models		X	X
Healthcare Models			X
Financial Fraud Detection			X
Security Analysis	X	X	
Biometrics		X	
Steganography	X	X	X
Neural Cryptography		X	
Model Privacy			X
Botnet Detection			X
Drive-By Download Attacks			X
Password Attacks	X	X	
Mobile Network Attacks	X		
Cracking Ciphers		X	
Vehicle Security			X
Universal IDS GAN		X	
GAN Models Discussed in Survey			
VanillaGAN	X	X	X
CGAN		X	X
DCGAN	X	X	X
WGAN		X	X
BiGAN	X	X	
CycleGAN	X	X	X
AC-GAN		X	X
ISGAN		X	
BEGAN			X
MsgGAN			X
ProGAN			X
SAGAN			X
IW-GAN		X	
InfoGAN			X
DefenceGAN		X	

overfitting[2] within the data. The FID calculated for any two distributions, μ and ν, over the set of real numbers, $\mathbb{R}^N$:

$$d_F(\mu, \nu) := \left(\mathit{inf}_{\gamma \in \Gamma(\mu,\nu)} \int_{\mathbb{R}^n \times \mathbb{R}^n} \parallel x - y \parallel^2 d\gamma(x, y) \right)^{1/2} \tag{7}$$

When we examine this equation it is of importance and interest to note that the set, $Gamma(\mu, \nu)$ is also called the 2-Wasserstein distance. It is possible to calculate the FID using a second method - but only under the specific instance in which the variable distributions are two Gaussian, multi-dimensional distributions, $\mathcal{N}(\mu, \sum)$ - symbolized below as r - and $\mathcal{N}(\mu', \sum')$, and this is therefore calculated as in Equation 8.

TABLE 2. Metrics in machine learning classifiers.

		Predicted Classification	
		Benign	Malicious
Actual Classification	Benign	TP	FP
	Malicious	FN	TN

$$FID(r, g) = \parallel \mu_r - \mu_g \parallel_2^2 + Tr(\sum_r + \sum_g - 2(\sum_r \sum_g)^{1/2}) \tag{8}$$

V. DATASET

This section briefly discusses the datasets used in the surveyed papers, their contents, type, and origins. While we

[2] Overfitting occurs when the data given is too specialized and the model fits itself too specifically to the given data, meaning that the generalizability of the model is lost, as is the ability to use it with future data.

did not want to devote too much time towards the datasets as opposed to the models, we felt it important to ensure that the reader had a solid foundation as to which dataset was being referred to and why it was appropriate for use.

A. NSL-KDD

In 1999, the KDD dataset was released as part of a championship game, The Third International Knowledge Discovery and Data Mining Tools Competition. The original purpose of the dataset and the competition was to have competitors building their own Network Intrusion Detection System [31]. This dataset was later refined and the problematic issues dealt with - the first issue was a large number of duplicate records which required removal from the set; the second issue was the way the data was structured, causing any IDS algorithm to achieve a 86% accuracy rate at minimum [32]. These issues were assessed and amended in [32], and the resulting dataset was dubbed the NSL-KDD dataset. This dataset of intrusion detection information has four categories: DoS, User to Root (U2R), Remote to Local (R2L), Probing Attacks. The training set contains 1,074,992 unique records: 812,814 are of benign traffic, and 262,178 are from the four classes of attacks listed above. The new and improved NSL-KDD test set contained 77,289 unique records. The updated dataset can be found on the website for the Canadian Institute for Cybersecurity at UNB, and is open access for researchers [33].

B. CIC-IDS

The CIC-IDS datasets are large sets of network traffic data. They include Benign data, multiple types of DoS (denial of service), DDoS (distributed denial of service), infiltration, SQL-injection, bots, port scans, and brute force attacks. Like the NSL-KDD dataset, the CIC-IDS datasets are available to researchers as an open source resource from the UNB Canadian Institute for Cybersecurity. It can also be found in numerous other locations across the web, as it is a popular dataset for training machine learning and IDS schemes. The name CIC-IDS is an acronym for Catalonia Independence Corpus Intrusion Detection System. There has been a significant body of research into the datasets from both 2017 and 2018, including a survey and taxonomy undertaken in [34].

1) CICIDS-17

The CICIDS-17 is the dataset created in 2017, using the data types and attacks most prevalent at the time. It contains real-time PCAP files of network traffic over the course of a work week - Monday 9am to Friday 5pm. In addition to the raw traffic flow files, it contains evaluations of the traffic data, and labeled and classified packets, with CSV files to deal with the network analysis information as part of the dataset.

The dataset's popularity has resulted in significant analysis. In [35], a thorough examination of the dataset was undertaken, with Feature Selection used to examine the 77 features in the dataset. They also utilized the processed data for machine learning as an effort to show in greater detail the effect this pre-processing data had on the training of a system.

TABLE 3. The distribution of class types over the raw CICIDS-2018 dataset. The imbalance shows clearly the need for preprocessing and data augmentation methods before using it to train machine learning classifiers [36].

Class Label	Count
Benign	2,856,035
BoT	286,191
Brute Force	513
DoS	1,289,544
Infiltration	93,063
SQL Injection	53
Total	4,525,399

2) CICIDS-18

Like the CICIDS-17 dataset, the 2018 updated version of the dataset contains real-time traffic files for analysis. The work done by [34] takes steps to examine the biases and imbalances of the dataset (as well as the earlier 2017 dataset). The assessment showed the skew of different data types, with the Benign data shown to be significantly greater in numbers than any other type, and some data types so small that the training of ML algorithms on the raw and full dataset would not create a balanced and effective IDS scheme. This can be seen in Figure 3, a table of the distribution of data types, or classes, in the 2018 edition of the dataset.

C. DARPA

The DARPA datasets date back to 1998 and 1999 respectively. They were considered early pioneers of data classes showing network attacks. The datasets were put together by MIT's Lincoln Laboratory, published in [37], with permission and involvement from the US government and Air Force. The full datasets can be found in places like Papers with Code [38], as well as on MIT's Lincoln Laboratory Research and Development website [37]. Similarly to the CICIDS datasets, the DARPA datasets contain the real-time traffic data, and an offline evaluation and assessment of the collected information. The DARPA datasets were collected based on attacks and daily traffic for an Air Force base. They include a large number of attack types. The data is separated as follows [37],

- Outside sniffing data (TCP dump format)
- Inside sniffing data (TCP dump format)
- BSM audit data (from Pascal)
- NT audit data (from Hume)
- Long listings of directory trees (from Pascal, Marx, Zeno, and Hume)
- Dumps of selected directories (from Pascal, Marx, Zeno, and Hume)
- A report of file system inode information (from Pascal)

Interestingly, in spite of the fact that the datasets are over two decades old, some recent papers have voiced their support of the DARPA datasets over the more recent (but still a decade old) NSL-KDD datasets [39].

D. CTU-13

The CTU-13 dataset is primarily used for training classifiers to recognize botnet attacks. It contains real-time botnet traffic of 13 different classes, and is one of the premier datasets for training ML IDS algorithms to recognize botnet traffic [40]. Due to the imbalance of classes in the CTU 13 dataset, there is a subset called the *Quasi-Balanced CTU-13 dataset* [41], which preserves the rare classes while balancing the number of instances with more heavily represented classes. The dataset has been used to validate results of training ML algorithms as in [42] and [43]. In both of the mentioned cases, the CTU 13 dataset was used to validate the results of training on the NSL-KDD dataset discussed in Section V-A. The dataset contains the following 15 features as part of the traffic flow captured for the dataset [40]:

- Start time
- Duration
- Protocol
- Source and destination IP addresses
- Direction
- Source and destination ports
- State
- Type of service (ToS)
- Total packets
- Total bytes
- Time comparison
- Average byte rate
- Average packet rate
- Ping byte
- Malicious port

E. DGArchive

The DGArchive is a set of domains, of 43 families, classes, or variants, with more than 20 million domains as of 2015 [44]. These domains are from models in Domain Generating Algorithms which create domains for Control and Command centers. The database of malicious botnet C&C domains allows for machine learning classifiers to be trained on how to detect domain name malware. This data is extremely important in creating new machine learning methods for identifying botnet C&C centers (as in [45]). The compilation of this information into such a large and comprehensive database is an important research tool. The DGArchive dataset is also used to create adversarial machine learning models, such as MaldomDetector [46], which undertake the generation of malicious domain names itself, and allows researchers to test defensive machine learning algorithms on an adversary.

F. RockYou

The RockYou dataset [47] is a comprehensive list of commonly used passwords, with more than 14 million entries, and has been used for brute-force dictionary attacks as well as for checking proposed passwords against. It is also used to train machine learning tools like PassGAN [48], which replaces the traditional password requirements which are chosen by a person, and creates its own requirements. PassGAN uses a Generative Adversarial Network to check and learn password distribution and such, and was trained on the RockYou dataset. It has also been used to train a Variational Autoencoder model for password guessing, in [49]. The dataset can be found in multiple locations, including Kaggle,[3] IEEE Data Port,[4] and TensorFlow,[5] among others.

G. ADFA

The Australian Defence Force Academy (ADFA) Intrusion Detection System dataset has two versions - a Linux version (ADFA-LD) and a Windows version (ADFA-WD). These can be downloaded directly from the UNSW Sydney[6] university website. The dataset was generated and compiled by Creech et al. over the course of three research papers, one of which was a doctoral thesis [50], [51], [52]. The dataset was originally developed to be used in research on Virtual Kernel Theory and capturing process calls, but has since been used for developing Intrusion Detection Systems (as in [53]), and in using k-nearest neighbor classification for cybersecurity (as in [40]).

H. UNSW NB15

This dataset from the University of New South Wales (UNSW) is an amalgamation of real traffic and generated attack data [54]. An intrusion detection dataset, it contains generated data from an IXIA traffic generator environment set up, The dataset is available on the UNSW's website,[7] as well as Papers with Code.[8] The dataset contains 2,540,044 instances, both benign and malicious [55], and was intended to be a successor to the NSL-KDD ('98 and '99 versions) and the DARPA IDS datasets [56]. Attacks are set in the following categories: Analysis, Backdoor, DoS, Exploits, Fuzzers, Generic, Reconnaissance, Shellcode and Worms. The breakdown of instances per class can be seen in Figure 4.

VI. TYPES OF GAN MODELS

We have surveyed papers containing a wide range of models of GAN schemes. From the base Goodfellow (or "Vanilla") GAN scheme to more complex version like the Wasserstein or the Conditional Deep Convolutional GAN (cDCGAN), these papers use models which are best suited to their needs. As a primer, or refresher for the more experienced researcher,

[3] https://www.kaggle.com/datasets/wjburns/common-password-list-rockyoutxt

[4] https://ieee-dataport.org/documents/rockyou

[5] https://www.tensorflow.org/datasets/catalog/rock_you

[6] https://research.unsw.edu.au/projects/adfa-ids-datasets

[7] https://research.unsw.edu.au/projects/unsw-nb15-dataset

[8] https://paperswithcode.com/dataset/unsw-nb15

TABLE 4. A breakdown of the instances per class in the UNSW NB15 dataset [55].

Class Label	Count
Benign	2,856,035
BoT	286,191
Brute Force	513
DoS	1,289,544
Infiltration	93,063
SQL Injection	53
Total	**4,525,399**

we have compiled the different types of GAN used in this literature for the ease of use of our readers.

A. GOODFELLOW GAN

The traditional, or Vanilla, Generative Adversarial Network, is that proposed in 2014 by Goodfellow et al. [1]. The traditional GAN follows the template set out in Figure 1. There are however, two important points to make with regards to this model.

1) MODE COLLAPSE

The problem of Mode Collapse - essentially an optimization problem - in GAN models is inherent to the *MinMax* game that is used to achieve optimal results. The model can fail because it has an inherently non-convex shape, making maximal values difficult to find with convex methods. Other models utilize different methods, for example the gradient descent-ascent (GDA) [57], to avoid the Mode Collapse.

2) CATASTROPHIC FORGETTING

The problem of Catastrophic Forgetting is one which occurs when the information gained in prior iterations of the model are lost or destroyed by the new task or iteration [58]. This is obviously a distinct problem because it makes it all but impossible to optimize the model as necessary, One of the outcomes of Catastrophic Forgetting is a failure to reach convergence. Both mode collapse and catastrophic forgetting are separate and interlinked problems - to fix one you need to fix the other [58]. There is discussion within the research community as to whether this issue is fixable utilizing Continuous Learning methods [59].

The structure of the Goodfellow GAN follows that of the general model sketched out in Section II. As such, we will not go into it deeply here. The Goodfellow model provided the structure on which these other methods were built. The optimal discriminator equation is shown in Equation 9, and the training for the Goodfellow Discriminator and Generator are in Equation 10.

$$D^*_G(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \tag{9}$$

$$V(G,D) = \int_x p_{data}(x) log(D(x))dx \tag{10}$$

$$+ \int_x p_z(z) log(1 - D(g(z)))dz \tag{11}$$

$$V(G,D) = \int_x p_{data}(x) log(D(x))dx \tag{12}$$

$$+ p_g(x) log(1 - D(x))dx \tag{13}$$

B. cGAN

cGAN, or Conditional GANs, as proposed in [7] by Mirza and Osindero, suggest a method for creating controls on the output of a GAN model, in order to create data samples with a focus on particular aspects. Mirza and Osindero discuss the benefits of being able to "direct" the process of GAN sample creation. For example, being able to focus the samples on the class labeling may be of use in some instances. In others, the focus could be on a certain feature in the samples. In this way, the cGAN model allows for an element of control that is lacking in the Goodfellow GAN. In [60], the authors discuss the ability the cGAN model creates to allow the researcher to employ different modes of operation for different tasks. The ability to focus on contextual input is a feature heavily discussed with respect to CGANs. While in a Vanilla GAN the structure in Figure 1, in a cGAN, the noise function $p_z(z)$ is combined with the conditional data, represented by y as input to the Generator. This is shown in Figure 2.

C. DCGAN

The Deep Convolutional Generative Network, or DCGAN, is a model put forward in [61]. They modeled the DCGAN architecture heavily on the original Convolutional Neural Networks that were used as building blocks for GANs. The original DCGANs were, as most GAN models were, originally heavily focused on image generation, learning, and classification tasks. These networks generalized well, and have since been successfully applied to security problems. Radford et al. incorporated multiple new techniques from several sources into their new GAN model (see [62], [63], [64]). These changes helped make it so successful in its tasks.

D. WGAN

The Wasserstein GAN was proposed in a 2017 conference paper [65]. The paper clearly lays out the two different distributions that are part of their contribution. The distance and divergences between our two separate distributions: $P_r, P_g \in Prog(X)$, in which $Prob(X)$ is "space of probability measures defined on X" [65, p. 215]. The distance between the two distributions is measured by the Total Variation (TV) distance, in Equation 14.

$$\delta(\mathbb{P}_r, \mathbb{P}_g) = \sup_{A \in \sum} \mid \mathbb{P}_r(A) - \mathbb{P}_g(A) \mid \tag{14}$$

The divergences between the distributions are measured with the Kullback-Leibler (KL) divergence (Equation 15), and the Jensen-Shannon (JS) divergence (Equation 16).

$$KL(\mathbb{P}_r \parallel \mathbb{P}_g) = \int \log(\frac{P_r(x)}{P_g(x)}) P_r(x) d\mu(x) \tag{15}$$

$$JS(\mathbb{P}_r, \mathbb{P}_g) = KL(\mathbb{P}_r \parallel \mathbb{P}_m) + KL(\mathbb{P}_g \parallel \mathbb{P}_m) \tag{16}$$

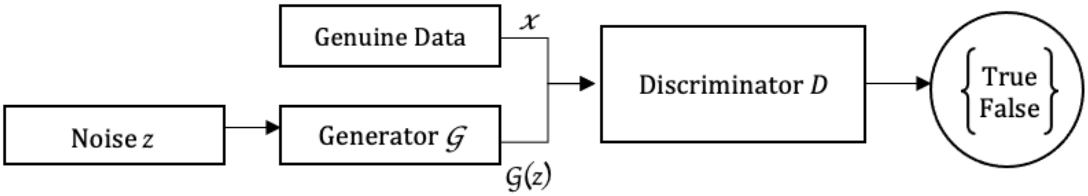

FIGURE 1. The simplified structure of a Vanilla GAN, as proposed by [1].

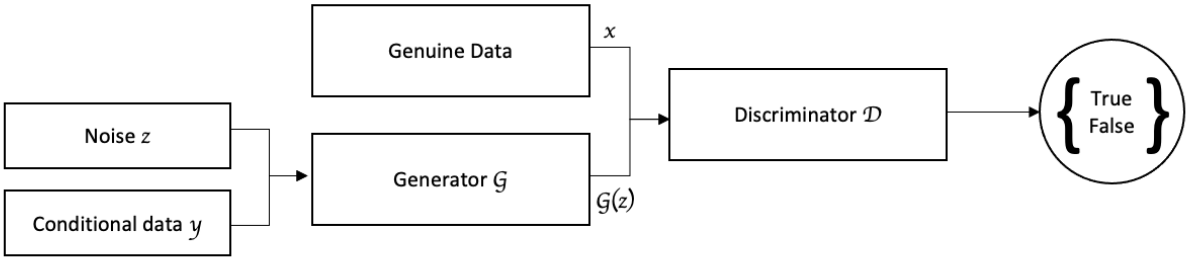

FIGURE 2. The structure of a conditional generative adversarial network based on that proposed by [7].

The central calculation to the WGAN is the Wasserstein distance, which is a part of the Earth-Mover equation, or the EM-distance. This equation tracks the distance between the result/outcome and the intended goal, rather than just a binary 0/1 evaluation from the classifier. This equation for the EM-distance is shown in Equation 17

$$WD(\mathbb{P}_r, \mathbb{P}_g) = \underset{\gamma \in \Gamma(\mathbb{P}_r, \mathbb{P}_g)}{inf} \mathbb{E}_{(x^r, x^g) \sim \gamma}[d(x^r, x^g)] \tag{17}$$

WGAN models have been widely adopted and used in many fields. In cybersecurity, they have been the basis of Intrusion Detection Systems, as in [66], in which the authors proposed a WGAN base for polymorphic adversarial cyber attacks, to train the IDS scheme against an ever-changing enemy.

E. BiGAN

The Bi-directional Generative Adversarial Network was proposed in 2017, by Donahue et al. [67]. The purpose of BiGAN models was to create a method of inverse mapping of the information backwards into the latent space. This inverse mapping offered more feedback to the network. It also created the ability for researchers to supervise learning with different focuses. The structure of the BiGAN model can be seen at its most basic level in Figure 3. The major change is the addition of the encoder to the two-party GAN model, creating a three-party game instead. While BiGAN models do excel in challenges in security, they also gained popularity in development for reading and generating diagnostic RNA predictions for the bio-informatic infosphere [68]. Others have already begun utilizing BiGANs in intrusion detection, such as [69].

F. GANG-MAM AND TrickDroid

The GAN used to create malicious Android apps, playfully named GANG-MAM, creates actual API calls for the purpose of compromising infected devices [70]. While only proposed for use in augmenting datasets and increasing the robustness of Android antivirus software, it is functional. This is not the only Android API malware creation to come out of the cascade of GAN development, and it shows the extent to which GAN can be used by an adversary for malicious purposes. In [71] the authors found that they could disrupt the accuracy of Android antivirus software based on machine learning systems by changing only 4 features of the 315 used for detection. They created a scheme to use this called TrickDroid, which created adversarial examples. The change of only 4 features dropped the accuracy/detection of those classifiers to 0%. This staggering piece of research showed how important a truly robust and tested system is needed in the realm of Android devices, and not just in traditional computer antivirus schemes. The finding is also dependent on the use of classifiers built using machine learning – this raises potential red flags about how ready these system are to be deployed and implemented for wide use. However, once the authors flipped the script and created a system to generate code injection attacks (CIA), the result was that the classifiers achieved under 1% in evasion rates. When they used their scheme, TrickDroid, to generate AEs for augmenting the dataset, the classifiers tested had their evasion rate dropped to 0.5% maximum across the board.

G. CycleGAN

The CycleGAN mechanism translates images from one domain to another [72]. There is more than just the one type of model - recent modifications include Mocycle-GAN, which translates video from one space into another [73]. The aims of CycleGAN models are to be able to take the input image from one domain and translate it into another domain or problem space belonging to the output sphere. A basic outline of the CycleGAN model is in Figure 4.

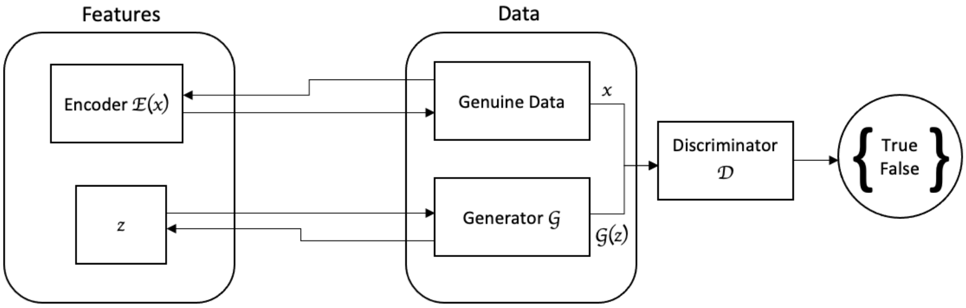

FIGURE 3. The structure of the basic Bi-directional GAN model proposed in [67].

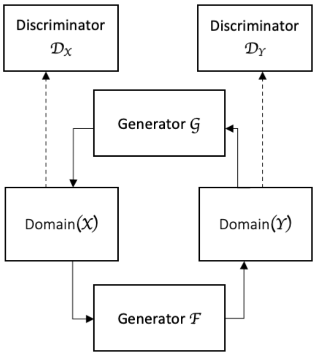

FIGURE 4. The CycleGAN model at its base structure for translating problem domains.

H. AC-GAN

The Auxiliary Classifier Generative Adversarial Network was proposed in [74]. It operates by increasing the structural requirements of the latent space of a traditional/vanilla GAN. They also added a cost function which was specific to their (image resolution based) task. The objective of Odena et al., was to characterize "the structure of natural images" [74]. The process involved down-sampling images to draw out the most necessary features. They utilized this to identify the point at which the ability to discriminate details within an image becomes an exceptionally difficult task. In [75], this model was moved into generating the more insidious malicious code attempting to attack the user system. They found this model to be particularly effective at this task.

I. PassGAN

In a 2019 paper, Hitaj et al. [48] proposed the GAN based PassGAN. This ML scheme was focused on learning likely password distributions from real lists, and creating its own password guesses. It was trained on the RockYou dataset of passwords (see Section V-F). The authors used the unique entries in the RockYou dataset for training purposes. In 2020, Biesner et al. [49] used a similar approach with Variational Autoencoders to create password guessing software trained via deep learning using multiple datasets, including the RockYou dataset discussed in Section V-F. The PassGAN algorithm was trialed against HashCat [76], a system to process and classify hashtags which has since been repurposed for many research areas, including through the creation of a "distributed hashcat" to harness these abilities [77]. The authors of [48] found PassGAN to be able to match 51-71% of passwords from the HashCat program. Being capable of undertaking this level of password generation anonymously is a difficult task. It also suggests that the use of GANs for attacking through password guessing has a high enough success rate that it will likely be an area of interest in not only research, but also in the development of new black hat techniques. As always, the balance of research and ethics is at play in situations such as this, and it is important to consider the potential misuse of any openly provided algorithms and how they are built.

J. IS GAN

The Identity Sensitive Generative Adversarial Network, introduced in [78], was proposed to generate sketches based on photographs. The reasoning for this was that the translation of the image often produced a great deal more detail than is noticeable in a photograph. The security applications of this model involve the ability to extract clearer images from CCTV images from crime scenes, among other things. While a very specific use-case is involved, it still offers an intriguing method of image-to-image translation for detail extraction.

K. BEGAN

The Boundary Equilibrium Generative Adversarial Network, or BEGAN, was proposed in a 2017 paper by Berthelot et al. [79]. The purpose of the BEGAN model was to employ the best of the WGAN model and the GANs that used trained autoencoders, while also changing the way that convergence was reached, so as to create a model that was fast and sturdy. The contributions discussed in the paper involve using a new equilibrium factor to balance the generator and discriminator networks; a method for sliding along the scale between diversity and quality; and the novel measure for *approximate* convergence. The new MaxMin, optimized objective equation is shown in Equation 18.

$$\begin{cases} \mathcal{L}_D = \mathcal{L}(x) - k_t \cdot \mathcal{L}(G(z_p)) & \text{for } \theta_D \\ \mathcal{L}_G = \mathcal{L}(G(z_G)) & \text{for } \theta_G \\ k_{t+1} = k_t + \lambda_k(\gamma\mathcal{L}(x) - \mathcal{L}(G(x))) & \text{for each training step } t \end{cases} \tag{18}$$

L. ProGAN

The Proximity Generative Adversarial Network was developed to preserve important semantic data - specifically, the proximity of data in the original space when downsampling, such that the proximity is preserved when the data is translated into a lower-dimensional space [80]. The ProGAN model not only preserves, but creates a method to generate proximities in sample data. The aim was to use this generation of proximity data to discover different semantic and characteristic traits in data with different proximities.

M. MSG-GAN

Multi-Scale Gradients for Generative Adversarial Networks, or Msg-GANs, are meant as an answer to the problems of domain transferability [81]. Because the gradients change specifically for the task at hand, taking an existing GAN model and modifying it for a new use is not a simple task. The idea of Msg-GANs is having multiple scales of gradients, which can pass from the discriminator to the generator. This also makes the system more stable.

N. SAGAN

The Self-Attention Generative Adversarial Model (SAGAN), was proposed in a 2019 paper by Zhang et al. [82]. This model was a variant with the specific distinction that the creator of the original/vanilla GAN, Ian Goodfellow, worked on the creation of SAGAN, for long-range image tasks. SAGAN models were created to allow the generation of details that come from a multitude of features, and a discriminator with the ability to check all these exceptionally detailed samples are consistent with one another. The addition of a ''self-attention'' module to the model offers the ability to calculate the full feature set and distances, returning a weighted sum with fairly little computational overhead cost. The main distinction of the SAGAN model is that it is essentially a convolutional GAN (see VI-B) with a self-attention module added. This module gives us the opportunity to add and define very fine details within images

O. IW-GAN

Inferential Wasserstein generative adversarial networks, or IW-GANs, melds Autoencoders and Generative Adversarial Networks together for greater functionality. Proposed in 2022 by Chen et al. [83], the IW-GAN employs a distinct and fast stopping criteria, and trains both the generator (G) and the deterministic autoencoder ($Q : X \to Z$) simultaneously. The results of their paper show IW-GAN as being effective at guarding against mode collapse. Rather than focusing on the Kullback-Leibler distance, the IW-GAN employs the 1-Wasserstein distance as its main evaluation tool. The equation for this can be seen in Equation 19.

$$W_1(P_X, P_{G(Z)}) = \underset{\pi \in \Pi(P_X, P_Z)}{inf} \mathbb{E}_{(X,Z)\sim\pi} \parallel X - G(Z)) \parallel \tag{19}$$

P. InfoGAN

The Information Maximising Generative Adversarial Network, or InfoGAN, model was first proposed in Chen et al. in 2016 [84]. In the paper in which InfoGAN was introduced, the authors noted the ability of InfoGAN's model to untangle images of handwritten characters. The model was tested and trained using the MNIST dataset. It was also utilized on 3-dimensional images of faces, and on pictures of house street numbers. In performance, the InfoGAN model adds ''negligible'' complexity to the vanilla GAN (see Section VI-A) model. The training itself was based on the training done for a DCGAN (Section VI-C), rather than a vanilla GAN.

Q. SeqGAN

A version of GAN developed for the purposes of generating data sequences, SeqGAN was proposed in 2016 by Yu et al. [85]. The SeqGAN model discards the generator differation problem and instead uses gradient policy the way we see in the common WGAN derivative, WGAN-GP. This was built from a need for a GAN model that could deal with sequences of discrete values, and not just binary, ''real-not-real'', continuous data. Using Gradient Penalty (GP) means that the generator can be coaxed bit by bit towards the goal, along a gradient path. These small but significant changes can be hard to undertake in the traditional continuous GAN model. Another reason why this varient is of interest is that the traditional GAN outputs a tally of real/not-real, and therefore would find giving a partial sequence as output difficult. To deal with this, the authors decided to classify generation of these sequences as a sequential decision-making process [85]. As part of using small changes along a gradient to alter the output of the generator, the authors propose a series of Monte Carlo calculations, and then train the generator using the policy gradient itself. The objective equation for the SeqGAN with GP is shown in Equation 20,

$$J(\theta) = \mathbb{E}[R_T \mid s_0, \theta] = \sum_{y_1 \in \gamma} G_\theta(y_{10}) \cdot Q_{D_\phi}^{G_\theta}(s_0, y_1) \tag{20}$$

R. TranGAN

TranGAN is the result of a transfer learning model whose purpose is to undertake social tie prediction [86]. This is an important piece of the puzzle for social network analysis. When tested against the traditional benchmark algorithms, TranGAN outperformed them and seems to have become a new standard in social tie prediction.

VII. AREAS OF USE

In the seminal paper introducing Generative Adversarial Networks, Ian Goodfellow states that the models' generators are ''analogous to a team of counterfeiters, trying to produce fake currency and use it without detection, while the discriminative model is analogous to the police, trying to detect the counterfeit currency'' [1].

TABLE 5. Taxonomy of papers reviewed.

Research by Area	
Adversarial Examples	Chauhan & Heydari [139] Duomoulin, Belghazi, Poole, Mastropietro, Lamb, Arjovsky, & Courville [100] Fang, Wang, Geng, Zhou, & Kan [140] Yan, Wang, Huang, Luo, & Yu [102] Zhao, Li, Wang, Zhang, Zhu, & Zhang [108] Zhu, Zhang, Yan, Chen, & Gao [146]
Autonomous and Connected Vehicles	Alheeti & McDonald-Maier [143] Cai, Wang, Zhang, Gruffke, & Schweppe [126] Kim, Kim, Jeong, Park, & Kim [125] Sedjelmaci [128] Seo, Song, & Kim [127]
Botnet Detection	Bansal & Mahapatra [43] Chowdhury, Khanzadeh, Akula, Zhang, Zhang, Medal, Marufuzzaman, & Bian [42] Velasco-Mata, González-Castro, Fernández, & Alegre [41]
Domain Generation Algorithms	Almashhadani, Kaiiali, Carlin, & Sezer [46] Choudhary, Sivaguru, Pereira, Yu, Nascimento, & Cock [45]
Image Translation	Cherepkov, Voynov, & Babenko [136] Choi, Choi, Kim, Ha, Kim, & Choo [134] Karras, Laine, & Aila [137] Ling, Kreiw, Li, Kim, Torralba, & Fidler Zhang, Xu, Li, Zhang, Wang, Huang, & Metaxas [135]
Intrusion Detection Systems	Creech [52] Creech & Hu [51] Draper-Gil, Lashkari, Mamun, & Ghorbani [106] Garuba, Liu, & Fraites [87] Khamis & Matrawy [54] Kulyadi, Mohandas, Kumar, Raman, & Vasan [90] Lee & Park [107] Lee & Park [104] Mouttaqi, Rachidi, & Assem [53] Park, Lee, Kim, Park, Kim, & Hong [103] Usama, Asim, Latif, Qadir, & Ala-Al-Fuqaha [91] Wang, Wang, Zhou, Li, & Zhang [105] Yang, Li, Liang, He, & Zhao [89]
IoT, Mobile, and Smart Grids	Abdalgawad, Sajun, Kaddoura, Zualkernan, & Aloul [110] Ferdowski & Saad [109] Grammatikis, Sarigiannidis, Efstathopoulos, & Panaousis [120] Umba, Abu-Mahfouz, Ramotsoela, & Hancke [119] Wei, Jiang, Yuan, & Wang [116] Zhang, Patras, & Haddadi [82] Zixu, Liyanage, & Gurusamy [113]
Malware	Amin, Shah, Sharif, Ali, Kim, & Anwar [138] Bae, Lee, Kim, Hwang, Yoon, & Paek [130] Bhaskara & Battacharyya [98] Choi, Shin, & Lee [92] Hu & Tan [132] Li, Kong, Xu, Qin, & He [147] Liu, Li, Liu, Gao, & Liu [93] Peng, Xian, Lu, & Lu [97] Tan & Truong-Huu [99] Smutz & Stavrou [129] Wang, Wang, Jiang, Wang, & Jing [94] Zhang, Wang, Sun, & Feng [131] Zhu, Zhang, Yan, Chen, & Gao [146]
Mode Collapse and Catastrophic Forgetting in GANs	Durall, Chatzimichailidis, Labus & Keuper [57] Seff, Beatson, Suo & Liu [59] Thanh-Tung & Tran [58]
Password Guessing	Biesner, Cvejoski, Georgiev, Sifa, & Krupicka [49] Hitaj, Gasti, Ateniese, & Perez-Cruz [48]
Privacy Preserving Models/Differential Privacy	Chen, Kairouz, & Rajagopal [20] Fredrikson, Lantz, Jha, Lin, Page, & Ristenpart [25] Hitaj, Ateniese, & Perez-Cruz [18] Huang, Kairouz, Chen, Sankar, & Rajagopal [21] Liu, Shiravastava, Du, & Zhong [22] Tripathy, Wang, & Ishwar [23]
State of the Art in GANs	Alrawashdeh & Goldsmith [24] Baluja & Fischer, 2017 [16] Haroon & Ali [34] Odena [141] Salehi, Chalechale, & Taghizadeh [27]

TABLE 5. *(Continued.)* Taxonomy of papers reviewed.

	Szegedy, Vanhoucke, Ioffe, Shlens, & Wojna [28] Zhao, Dua, & Singh [19]
GAN Development and Design	
Auxiliary Classifier GANs	Odena, Olah, & Shlens [74] Nagaraju & Stamp [75]
BeGAN	Berthelot, Schumm, & Metz [79]
Bidirectional-GANs	Renjith, Laudanna, Aji, Visaggio & Vinod [70] Rafiq, Aslam, Isaac, & Randhawa [71] Xu, Jang-Jaccard, Liu, & Sabrina [88] Xu, Jang-Jaccard, Liu, & Sabrina [69] Yang & Li [68]
Convolutional GANs	Gauthier [60] Ioffe & Szegedy [64] Radford, Metz, & Chintala [61] Springenberg, Dosovitskiy, Brox, & Riedmiller [62]
CycleGANs	Amsaleg, Huet, Larson, Gravier, Hung, Ngo, Ooi, Chen, Pan, Yao, Tian, & Mei [73] Zhu, Gong, Qian, & Zhang [72]
IsGAN	Yan, Zheng, Gou, & Wang [78]
InfoGAN	Chen, Duan, Houthooft, Schulman, Sutskever, & Abbeel [84]
MSG-GAN	Karnewar & Wang [81]
ProGAN	Gao, Pei, & Huang [80]
RenderGAN	Sixt, Wild, & Landgraf [145]
SeqGAN	Yu, Zhang, Wang, & Yu [85]
TranGAN	Chen, Xiong, Liu, & Yin [86]
Wasserstein GANs	Arjovsky, Chintala, & Bottou [65] Chauhan, Sabeel, Isaddoost, & Heydari [66] Chen, Gao, & Wang [83] Donahue, Krähenbühl, & Darrell [67] Wang & Wang [105]
ZipNET-GAN	Zhang, Ouyang, & Patras [118]

This adversarial model is what makes GANs so excellent in many areas. In this section, we will discuss the areas in which GAN models have been most successful, with a particular focus on those relevant to the creation, training, and maintenance of Intrusion Detection Systems. Intrusion Detection Systems default into several main categories. For the purposes of this paper, and the review of IDS experimentation with GANs, we have sorted them into the following categories: Wired or general Network IDS; Wireless; IoT; Mobile; Sensor Networks; and Autonomous Vehicles. These are the main types of Network Intrusion Detection Systems, and the main focus of this paper. Traditional IDS methods involve anomaly detection and attack signatures, with specific definitions for *what* the scheme should be looking for [87]. This section describes the different areas in which Generative Adversarial Networks are most useful in assisting a Network Intrusion Detection System. The use of GAN models to train intrusion detection systems, or IDS, is a fundamental use-case in cybersecurity. Between the ability to generate new examples, create adversarial application files or traffic, and highlight the important contextual clues and relationships, GAN models have significant contributions to make in the training of new IDS schemes [88].

A. NETWORK INTRUSION DETECTION SYSTEMS

In traditional Network IDS machine learning models, GANs are used in multiple aspects to improve performance. For example, in [89] the authors take advantage of the strengths of Generative models, using the Deep Convolutional Generative Adversarial Network (DCGAN) and Long Short-Term Memory (LSTM) methods to design an effective real-time intrusion detection system for use in general devices. The DCGAN is specifically chosen to help balance out the positive and negative samples by generating new synthetic, raw data. As stated in Section VI-C, the DCGAN is excellent at generalizing, and has been applied to multiple security problems, including [61]. The LSTM then provides the classification method. This proved highly effective, and when tested against the KDD and NSL-KDD datasets (see V-A), was able to achieve 99.73% and 99.62% accuracy respectively. In [90], the authors use a GAN scheme to learn the patterns in their traffic log data, training the model to recognize the types of traffic, and then using this to detect any anomalies in the traffic patterns. This creates a GAN-based system for detecting malicious traffic. Their model achieved an f1-score of over 94% when identifying the anomalous traffic.

Some researchers have been using GANs in creative ways to improve network security, for example, in [91], authors attempted to create a pair of GAN schemes - one to attack and one to defend. They used a GAN-based IDS for the detection of attack data, and to defend against it. While their overall accuracy numbers were not as high as one might hope, they did show that it is possible to use GAN-based schemes to defend against the types of attacks leveraged against a machine learning or deep learning based IDS model. Using a GAN to create the adversarial examples and a second GAN to detect and defend against said examples is a creative approach to two-party security models. This is an area of research with great potential.

There have been many interesting, suggested models for GANs to run on, including one which suggested that pulling the opcodes, the machine instructions, from the program, with the purpose of comparing byte sequences with known malware examples, may offer high level accuracy in identifying the variant [92]. This approach offered some interesting possibilities. The scheme focused primarily on the protection of high-security systems, like weapons or defensive programs. This is an area of urgency when it comes to accurate detection of malicious traffic and software. The authors proposed that one might use opcode sentences, sequential strings of the machine instructions, for the classification and the generation of new sentences. The scheme resulted in a significant improvement in detection accuracy, jumping from 96.3%, to 98% when the GAN-augmented data was added to the training set. This was with an experimental setup with such limited data, the adversarial samples from the original dataset numbered only 42. Further tests showed the area under the curve (AUC) went from 79.2% to 98% when the augmented dataset was applied to the training of the model. This clearly displays the success that is possible when using GAN models to generate adversarial examples, even on the rarest classes. In [93], similarly to [94], the authors implement a GAN in order to classify malware samples through translation to images for feeding into the GAN scheme. The Mal-IAGAN model they propose also trains IDS models using the classified images. The significant contribution they make in this paper is the robustness of the solution. Even when the Mal-IAGAN is only trained on 1% of the dataset, an amalgam of VirusShare APK Android malware [95] and the BIG-2015 dataset [96], the model had an accuracy rate of over 80%. This suggests the model has an excellent robustness with regards to unseen examples, and that the model can generalize to a significant degree.

Reference [97] focuses specifically on the use of API calls within Windows executable files for the identification of malicious code. The authors use a GAN model to train their own classifiers, both of which achieve impressive results in identifying malware samples. The contextual and semantic relationships are essential to identifying the malware through the API calls it makes. The authors use a Long Short-Term Memory (LSTM) model GAN, and as their classifiers they utilize models they name LSTM-Attention and BiLSTM-Attention. The proposed models are measured against several existing machine learning classifiers for their performance as IDS models. All of them are trained using their GAN scheme. The comparison of Convolutional Neural Network (CNN), Logistic Regression, Decision Tree, Random Forest, Support Vector Machines, and Multi-Layer Perceptron models show excellent performance, with 95.43% and 96.53% accuracy on the LSTM-Attention and BiLSTM-Attention respectively.

In [98], the authors propose a method to use GAN models to train their IDS using RGB images of malware for classification purposes. The authors wanted a way to continually update and train their antivirus software, after it had been released. Using GAN models offered the opportunity to continue providing new formations of malware and unseen examples to train their software. This ‘‘update and retrain’’ behavior, also called online learning, is present in [99], in which the authors claim that they can use GAN models to deal with the issues presented by the deterioration of machine learning models over time. The authors used multiple GAN models - DCGAN [61], ALI-GAN [100], CoRGAN, and CoRaGAN. CoRGAN and CoRaGAN (created by the authors themselves), and DCGAN, ALI, and CoRGAN consistently perform at the top of the different metrics and databases. The highest scores in precision, recall, f1-score, and accuracy were all in the high 90%, and the augmented dataset improved the scores across the board.

In [101], the authors propose a combination network which utilizes both Convolutional Neural Networks and WGANs to create an IDS system to detect and classify threats to the system. The use of a WGAN (discussed in Section VI-D) is primarily aimed at improving model stability and minimizing the chances of mode collapse. While they achieved a high rate of accuracy on the test set, there were also 17 classes of attacks which were not seen in the training set but were included in the test set. On these unseen attack samples the system achieved an impressive 67.5% accuracy rate in classification. The accuracy in classifying the binary experiments was 88.23% and the accuracy in classifying the five main classes was 80.80%. The ability to correctly classify unseen classes shows exactly how powerful these models can be. While 67.5% is certainly lower than one would expect to achieve on classes the model was trained on, it is significantly higher than the expectation for classes that the model has never seen, which in this case would have offered a random chance of at most 1/17. In [102], the authors utilize a WGAN derived method called DoS-WGAN, specifically to generate new samples of trace evidence from DoS attacks for the purpose of training IDS schemes to detect these types of attacks. The DoS-GAN allows the camouflage of attack traffic, while the Standardized Euclidean distance and the information entropy are used to measure progress in training. The authors are specifically focused on the importance of examining how attackers are most likely to adapt to the knowledge that the system they are trying to compromise is utilizing a machine learning based IDS defense mechanism. This focus is shown in the ways that the authors utilize their DoS-GAN method to perturb and manipulate the malicious samples for detection evasion. This paper is specifically focused on the attack side of the IDS research question, using the DoS-GAN model to attack and evade existing ML IDS models. Their success in this shows the ways GANs can be used not only for, but against IDS models. In [103], the authors focus on the reconstruction error and the Wasserstein distance while creating an AI based NIDS scheme which utilizes both GAN and autoencoder methodologies. Three machine-learning classifiers were used: deep neural networks (DNN); convolutional neural networks (CNN); and Long Short-Term Memory (LSTM) models. The experimental set up was tested on the NSL-KDD (both versions), UNSW-NB15, IoT datasets, as well as a ''real-world'' dataset of normal/benign network traffic. A Support Vector Machine (SVM) and Decision Tree (DT) were employed as comparative models for the experiments. The proposed GAN NIDS scheme achieved scores of 93.2% and 87% on the NSL-KDD and UNSW-NB15 datasets respectively. In several categories on the IoT dataset the model achieved accuracy of 100%. This robust and competitive performance showcases the effectiveness of GAN based schemes for network intrusion detection systems. Similarly, in [104] the authors employ GAN and a Random Forest model to examine and detect attacks in network traffic, only using the CICIDS 2017 dataset. The use of GAN methods in conjunction with the Random Forest classifier resulted in high results across the board, and they were compared to the results of a single RF classifier, with the accuracy, precision, recall, f1-score of the GAN RF model achieving 99.83%, 98.68%, 92.76%, and 95.04% in comparison to the single RF model's scores of 99.19%, 98.2%, 83.79%, and 87.79% respectively. This again emphasizes the utility and strength of GAN models in creating robust IDS schemes.

B. WIRELESS NETWORK INTRUSION

Intrusion Detection systems that reside on the Network layer of communication infrastructure for distributed schemes rely on a robust security level to secure communications between devices. This is an essential part of securing any business or government network. Any connected system of devices that uses internet connectivity relies on NIDS models to remain safe and to enforce the CIA principles of security. One such example, using PCAP files for training and testing, called FlowGAN, sets about doing exactly this [105]. This method improves the accuracy of identifying malicious network traffic significantly, and the authors utilize a dataset introduced in [106], called ''ISCX VPN non-VPN traffic dataset'', for the experimentation portion of their study. The Precision, Recall, and F1-Scores were increased by 13.2%, 17%, and 15.6% respectively, when run against the same algorithms using a dataset unedited by the FlowGAN model, using a Multi-layer Perceptron model in both cases. The ability to use the model on both encrypted and non-encrypted traffic shows its usefulness. Many businesses and other connected groups rely on connections that run through VPNs, meaning a model like FlowGAN being capable of operating over encrypted traffic is extremely useful in real-world scenarios. In [107], the authors employ an Autoencoder Conditional GAN (AE-CGAN) model to improve intrusion detection on the network, using the CICIDS 2017 dataset (see Section V-B). The authors compared this model against two others - single RF, and AE-RF - and found that the proposed AE-CGAN model showed improved accuracy in comparison with the other two. They made note of the importance of feature extraction in identifying malicious network traffic through an IDS. The use of an autoencoder for this purpose allows the IDS model to continually modify itself and adapt to environmental changes within the network, using unsupervised learning. In the 2022 review paper on Adversarial Machine Learning methods for securing wireless and mobile networks, the authors [22] explored the current state of GAN research in the area of wireless networks and the relevant intrusion detection systems. This is a very thorough survey of the state-of-the-art in the area, and the inclusion of GAN models makes it particularly relevant to the work presented here. The authors note that GANs generally require access to all the features, including the functional and non-functional. This is because of the need to generate realistic data that approximates the genuine article in all ways, and is therefore a core requirement of the process of training a GAN model. They also particularly highlight the use of GANs in creating adversarial examples, both for attack and for training purposes.

C. INTERNET OF THING INTRUSION

In [108], a method referred to as attackGAN is used to build attacks that take advantage of the weakness of machine learning models. They use their model to attack the perturbation of data on IoT devices. This method is utilized to demonstrate the deficiencies of the current methods and ways they can be improved. The attackGAN model is based on the previously discussed Wasserstein GAN model (Section VI-D), with feedback from the IDS scheme used to improve later attacks. The authors also made use of the NSL-KDD dataset for the development of the GAN model (see Section V-A for details on this dataset). Using GANs for adversarial examples like this is an excellent option for training an IDS to react appropriately to zero days or unseen classes of attacks. GANs offer more generalization in augmented datasets, which helps prevent overfitting when training the ML IDS model. IoT devices can be used in concert to create distributed systems. Ferdowsi and Sand [109] do exactly this, in creating a distributed GAN-based IDS for IoT systems. Their model achieved an accuracy of up to 20% higher than a non-distributed IDS method. Because they distribute the system over all the different IoT devices (IoTD) on the network, the system also provides an option for creating more stable IDS methods in networks with resistance to the failure of individual devices. Each individual device is optimized for detection using the value function in Eq. 21.

$$V_i(\bar{D}_i, \bar{G}_i) = -\log(4) + s(p_{data_i}||p_{data}) \tag{21}$$

In [110], the authors use the newly published IoT-23 [111] dataset and methods such as Bi-directional GAN, or BiGAN (see Section VI-E), to train IDS models to detect attacks like those from the Mirai botnet, which at its peak infected more than 600,000 IoT devices [112]. The IoT-23 dataset involves the network traffic records of devices such as smart-doorbells and Amazon's Echo smart home hub, and is composed of log files generated from .pcap files with labels generated through use of a python script, thus avoiding the time-intensive requirement of individually labeling the samples by hand. The authors were able to use their models to achieve an impressive F1-score of 99%. BiGAN models, as discussed earlier, are specifically for the purpose of allowing inverse mappings and the ability to specify focuses. Their BiGAN model for the detection of zero-day and unseen attacks achieved an F1-score of between 85% and 100% over the different classes of the data. In [113], the authors combined a Wasserstein GAN and an Autoencoder for the creation of an IoT network IDS scheme which also uses the Gradient Penalty scheme to improve performance. They used the Bot-IoT dataset from the University of New South Wales [114] for training and testing purposes, and identified within that dataset of traffic flows 9 main features on which to base their training - 2 categorical features and 7 statistical features. Categorical features are run through one hot encoding systems to prepare for use, resulting in a dimensionality of 29. Features are also normalized prior to their use, ensuring ranges are kept to [−1,1]. As part of the experiments, the authors trained both a Global Model, and a Distributed Model. The Global Model is a single instance of the scheme with access to all local samples and data. The Distributed Model, on the other hand, involves giving each local network its own local autoencoder, trained only on the local data and samples, and not linked to the other instances. The overall performance was compared using four different clustering methods: one-class support vector machine, isolation forest, local outlier factor, and K-Means clustering. The traditional metrics of precision, recall, accuracy, and F1-Score were used to measure the resulting performances. The overall best performer was the Global Model, with accuracy, precision, recall, and F1-scores of 97.11%, 99.33%, 97.33%, and 98.31%, respectively. This shows there is a space to utilize GAN-based NIDS for distributed IoT systems with a high degree of confidence and a significant success rate. Further research in this area is needed, and this is a potential research space with the opportunity for serious real-world applications.

D. MOBILE INTRUSION

Given the prevalence of mobile devices in the current technological era, the ability to secure these devices is of exceptional importance. Mobile devices contain scores of personally identifiable information (PII), as well as being the portal by which we see the world. As stated simply in [115], ''the more widely a technology is used, the more likely it is to become the target of hackers''. In [116], the author employs a Wasserstein GAN model to develop a malware detection system for mobile systems. This scheme is specifically for detecting suspicious behavior and communication on the network layer of a mobile device and could therefore also be considered a form of Network IDS. They used 559 applications from the Android Play Store, from a large variety of categories including entertainment, news, system tools, etc. The test set for the WGAN model found the accuracy of detecting malicious network behavior to be approximately 88%. When the author included generated data in the sample set the accuracy improved to 96.89%, demonstrating that a GAN model can even create application files that are able to act in place of genuine sample files. This is not an insignificant finding.

In [117], the authors examine the research into newer advanced machine learning methods and mobile and wireless networks. They touch on the use of GANs for data generation, particularly for supervised learning tasks. While it does not focus specifically on security measures, there is discussion of the different ways in which GANs were in use for mobile network analysis training and Mobile Traffic Super-Resolution. GANs are particularly successful at this task, with their initial aim of image generation being translated into adversarial examples in many papers.

Utilizing GAN models to develop a secure mobile network, in [118], the authors combine a GAN model with Zipper Network (ZipNet). The goal in this paper is to create a system that can deal with the large scale requirements of mobile traffic analysis city wide. Their scheme can infer details with up to 100 times the granularity of standard probing methods. It was potentially the first time a system has employed super-resolution methodology to mobile traffic analysis. The scheme results in between 65 and 78% smaller Normalized Root Mean Square Error, or NRMSE. There is certainly scope to undergo further research in this area, as the security of the mobile network from intrusions and malicious traffic is of vital importance with the proliferation of mobile technology.

E. SENSOR NETWORK INTRUSION

Sensor networks, like those in smart grids, are an aspect of IoT devices large enough to require their own section in this paper. Given their use in areas such as public transport, power plants, medical devices, and other areas of national infrastructure, the security of these devices is of national importance. In [119], the authors review the current (as of 2019) methods in use for machine learning based intrusion detection systems in wireless sensor networks. The Software-Defined Wireless Sensor Networks, or SDWSN, are a combination of Software-Defined Networks and Wireless Sensor Networks. Software-Defined Networks are found across medical and industrial devices, as well as in the use and guidance of drones and bombs. As such, they are high-value targets in need of robust IDS methods. The possibility of a malicious actor hijacking one of these devices or networks is far too serious a threat to ignore. Reviewing the state-of-the-art in protecting these devices and their networks, the authors found that combining machine-learning or AI methods with cryptographic schemes to be the most effective way of securing the SDWSNs. GANs were taken here as effective methods of augmenting and improving the datasets for training these ML/AI intrusion detection systems. In [120], the authors developed a new GAN based intrusion detection system for Smart Grid networks. The scheme, called ARIES, utilizes 3 different detection layers for maximum protection. It scans and covers network flows, Modbus and transmission control systems, and the operational data. Utility grids and energy companies in most Western countries are considered to be Critical National Infrastructure, and therefore require a high level of security [121]. Attacks against CNI can be disastrous for the people within a country, and thus any intrusion into the networks that control and maintain CNI must be detected and dealt with as soon as is possible. Smart Grids, a type of sensor network that deals in the maintenance and visibility of an energy grid, are highly connected networks, and therefore require sophisticated cybersecurity systems. The ARIES GAN system involved the use of electrical signal increases from a power plant in Greece to detect control commands and abnormalities, the first to do so. This information was collected as part of the operational data layer. The CSE-CIC-IDS2018 dataset [122] was used for the testing and training of the network. This dataset includes network flow statistics and other control data which was combined with data from the Greek power plant for specificity of information. Using a Decision Tree classifier resulted in the best scores in the first detection layer (IDM) for accuracy, true positive rate, false positive rate, and f1-score, being 99.4%, 98.2%, 0.3%, and 98.2% respectively. In the second detection layer the best results were found with an Isolation Forest classifier at 91.7%, 75.1%, 4.9%, and 75.1%, while the third detection layer was best served by the ARIES GAN system at 93%, 87.5%, 5.3%, and 85.3% respectively. These levels of accuracy show the potential for an ML IDS to protect CNI sensor networks. As the use of smart sensors in CNI systems grows, so does the need for truly secure IDS models. Therefore, there is a need for a concerted research effort in this area, and GAN models seem likely to offer significant improvements.

F. AUTONOMOUS VEHICLE INTRUSION

Autonomous vehicles, like Sensor Networks, are technically an IoT subsection. However, they are similarly prevalent and serious enough to require their own addition in this paper. Plenty of research in recent years has focused on the development and use of autonomous vehicles, known colloquially as self-driving cars. Because these machines are usually in constant communication with the cloud-based services that provide their data and instructions, it is of extreme importance to secure them against intrusion. Hacking cars, even traditional vehicles, has been shown to be both possible and effective. As early as 2015, Wired published an article describing the way two researchers remotely hacked a Chrysler vehicle, prompting a massive recall of over 1.4 million Chrysler vehicles [123], [124]. Given the increase in connectivity from traditional to autonomous vehicles, the security of these devices is a life-or-death situation. As such, researchers have begun to seriously examine the security concerns of malicious intrusion into autonomous vehicles. When reviewing the current state of the art in security for AVs, the authors of [125] gave a comprehensive review of cybersecurity for vehicles. They were careful to highlight the important security flaws found by Keen Labs in Tesla vehicles in 2017,[9] followed by BMWs in 2019 [126], and the newer security risks posed by the popularization of autonomous vehicles, which depend heavily on the ability to reach and communicate with global servers for updates and information on routes, conditions, and traffic alerts. BMW was the target of security vulnerabilities in [126] where the authors described the exploits and attacks found using the Infomatic systems and the networked entertainment modules. These systems - for which the vulnerabilities were addressed by BMW prior to the publication of the paper (an example of the success of researchers ensuring ethical publication of security research) - allowed the researchers to access the on-board computing modules and deploy commands to the vehicles. Researchers in [125] also found that the majority of the research surveyed displayed a tendency towards using machine learning and artificial intelligence methods to secure these new vehicles. This security need created by the rise of autonomous vehicles is one that machine learning researchers have begun exploring, leaving opportunities for research into the potential use of GAN models to create secure network intrusion detection systems for these vehicles. One example of this can be seen in [127], where the authors proposed a GAN-based Intrusion Detection System they named GIDS. The focus of this system was on effectiveness, expandability, and security. Because the training was exclusively performed on normal data, the system could detect intrusions and attacks without focusing on a particular type of attack data. The idea of this type of training was that the IDS would be able to better detect unseen attacks this way. The authors exploited the image-based excellence of GANs by converting CAN messages into images for use in the system, in a process referred to as "one hot-vector encoding". The network used to classify was a combination of Convolutional Neural Network and Deep Neural Network. The authors tested the system with DoS, Fuzzy, and RPM/GEAR attacks, as well as benign or "normal" data. While the larger sized inputs did decrease overall accuracy (with the most significant dip at 80) the input size defined for the final experiments was fixed at 64. The lowest detection rate for an input data type was RPM attacks, at 98.7%. It was able to operate in real time, as it took 0.18 seconds to sort 1,954 CAN bus messages, and in practice the CAN bus system generates approximately 1,954 messages per second. This very effectively demonstrated the potential impact of a GAN based system for creating an effective IDS for vehicular systems, but there is still plenty of research space in this area.

In [128], a review of IoT NIDS and machine learning systems, the authors put forward the use case of autonomous vehicles and the vehicular edge network as the example

[9] Keen Security Lab of Tencent, https://keenlab.tencent.com/en/2017/07/27/New-Car-Hacking-Research-2017-Remote-Attack-Tesla-Motors-Again/New Car Hacking Research: 2017, Remote Attack Tesla Motors Again, 2017-07-27.

of GAN and NIDS in networked devices. The processes undertaken as part of the vehicular edge network are carried out on the Mobile Edge Computing server, or MEC. When a vehicle needs to undertake a process that can be done faster on the MEC server than on its own computational equipment, the network offloads the process to the MEC. The vehicular edge computing system responsible for this division of labor, or VEC, is a 5G network connecting the vehicles to the MEC for secure communication. Of course, as with any external network connection, it is vulnerable to attack. The security system proposed by the authors suggests the embedding of the GAN based scheme at each of the nodes, monitoring any traffic to or from the MEC server. The security scheme is monitored by each MEC node, meaning that the MEC servers themselves are able to detect and react to malicious activity within their network sector, as well as allowing them a global view of the network and its security. While the authors were fairly non-specific about the types of GAN algorithms employed to work on the MEC servers, or the general set-up and use, they did specify that they were able to achieve an accuracy rate of up to 90%.

VIII. DISCUSSION

The previous sections have primarily set the stage for this discussion - why, how, and where is it appropriate to employ a GAN model for the improvement of Intrusion Detection Systems? The importance of discussing where *not* to use GAN is as important as discussing the ways in which GAN is being effectively employed.

A. WHY GAN?

Goodfellow asserts that the two-player game, with the heavy intervention of backpropagation methods, is what makes Generative Adversarial Networks so effective in their tasks. The derivatives used for that backpropagation are calculated as seen in Equation 22.

$$\lim_{\sigma \to 0} \nabla_x \mathbb{E}_{\epsilon \sim \mathcal{N}(0,\sigma^2 \mathbf{I})} f(x+\epsilon) = \nabla_x f(x) \tag{22}$$

1) WHEN TO USE GAN

The Generative Adversarial Network model has specific traits which make it better at some specific tasks than others. We have explained the ease with which GAN undertakes image-based tasks below (see Section VIII-A2). However, this does not mean that the use of GAN schemes is restricted to those which are naturally image-based. Many tasks can be translated into image domains, or can be fitted with the data they have, like those we have seen use the opcode sequences [71], or PDF files [129], [130], or even APKs [131] and API calls. In the training of an Intrusion Detection System, the generation of Adversarial Examples [132] is of exceptional importance. Creating samples to "attack" the system to train it offers the ability to train it in a semantic manner with contextual clues. This can offer strength when the IDS is faced with zero-day attacks, as it relies not only on previously seen training samples. The use of generated Adversarial Examples can offer an improvement on generalization and learning traits from families of malicious code. It can also help protect against overfitting, especially when only small numbers of samples for a particular class are available for training a network. In the case of IDS models, having an attack/defend scheme, such as the one discussed in [91] (see Section VII-A), offers the ability to view real-time reactions from the defender network in a controlled environment. Building an attack model like this creates opportunities to test the IDS model in a controlled environment in real time, which can be invaluable in debugging and streamlining the system.

In an example like MalGAN [132], the GAN model is used to create malware for the purposes of training and testing Intrusion Detection Systems which are based on machine learning methods. This is a key point - IDS models built through machine learning methods can be a good counterpoint to use GAN attack methods on. However, traditional IDS models are unlikely to gain much through the use of a GAN attack model.

Adversarial examples are of course, not the only area for employing GANs for use in IDS models. Generative models can be used for creation and classification in many ways. The discussed areas of Sensor Networks and Autonomous Vehicles are perhaps the most important or essential areas of research when it comes to machine learning IDS models, and thus are an area for focusing GAN research.

2) HOW TO USE GAN

Generative Adversarial Networks are highly useful models for many tasks, when implemented correctly. While this paper is specific to Intrusion Detection Systems, the methods of implementing GANs are standard across many research areas. However, the framework for using GAN models requires researchers to decide on tasks with care, so as to implement GAN models when they will be most useful. We have iterated some of the tasks in which GAN methods are most likely to provide useful output, with discussion of how and why they work in the mentioned tasks.

a: DATA PRE-PROCESSING

GAN models are excellent at tasks that involve balancing datasets or sampling rare data classes for training and testing of other Machine Learning classifiers. These tasks often break down into image-based samples, and non-image samples, because of GAN's ease of use in the image domain. In the realm of ML based IDS models, balancing the rarer attack classes in datasets is an extremely important part of training an IDS method. In some datasets there are as few as a couple of dozen samples of a specific class. Traditional methods like SMOTE or ADASYN may not be able to augment these classes without creating overfitting, which is what makes GANs so useful.

1) Image Samples
   Image tasks are an area GAN models are extremely competent in, with computer vision, imaging, and

other domains well-saturated with GAN based schemes [133]. One excellent example is StarGAN [134], which the authors trained to take facial images, using celebrities for training and testing, and translate them into different hair colors, genders, emotions (such as happy, angry, and fearful), and skin colors. GANs are regularly used in tasks that involve image-to-image translation, text-to-photo translation [135], and image generation.

2) Non-Image Samples
GANs may work particularly well on image based tasks, but they are also of great use in tasks that involve samples from non-image domains. While it is possible to translate a non-image data type into an image for ease of processing (see below), it is not always necessary.

3) Changing to an Image Domain
As we have seen throughout this paper, GAN models can be trained on data that has been translated from a non-image sample to an image sample. The translation of traffic flow, PCAP file, application files, and executables into images allows IDS researchers to take advantage of the strength of GANs' image classification abilities. There are a number of methods for translating data to image, such as [134], [136], and [137]. In particular, the translation of .PCAP files, applications, and other data types into an image for ease of operation is quite common among researchers in the cybersecurity domain, due to the general success that GAN models have with image-based tasks. This enables security researchers to maximize the performance of their GAN model for IDS while using traditional IDS datasets with non-image data.

4) Non-Image Sample Types
In more recent years, as GAN models have proliferated from the computer vision discipline into countless other subject areas, including cybersecurity, researchers have increasingly employed GAN methodology with non-image data types. Within this paper we have explored research that dealt with opcodes [131], APKs [138], network flow traffic [118], and many other data types. Papers such as [89] have used network data of attacks such as the KDD and NSL datasets for training and testing purposes, showing how versatile these methods are. For IDS researchers, the ability to use untranslated datasets saves significant time in the pre-processing stage, as well as computational power. Generally, IDS datasets for research do not appear in an image format, so the ability to use a GAN without translating the data to an image first is of great importance in training and testing models for intrusion detection. It also enables more realistic opportunities for real-time operation, as the time taken to translate incoming data to images in order to classify it could significantly increase processing time.

b: ADVERSARIAL EXAMPLES, UNSEEN ATTACKS, AND ZERO-DAY SAMPLES

Throughout Section VII, we have demonstrated the effectiveness of GANs in creating attacks and adversarial examples. For example, in [139], the authors present a GAN-based method for continuously changing the attack profile of a system so that it remains undetected by the IDS. The focus of the paper is on polymorphic attacks, those which are constantly changing in order to remain under the radar. Using GANs to create polymorphic attack data shows the versatility with which these systems produce synthetic samples. They used the GAN models to swap different features of the benign data samples with features from the malware samples it was trained on, to introduce characteristics of the benign data into the adversarial examples. This type of attack method is extremely difficult to counter, and offers a serious risk to those developing traditional IDS models. Using a Random Forest classifier to test the effectiveness of their model, the authors found that after 100 epochs and having swapped features, they were able to achieve a detection rate as low as 3.89%. This achievement shows the impact that GANs can have when used to create adversarial examples to evade IDS models. It also opens the doors to more research into how best to counter these attacks when deployed in real-world scenarios. In [140], the authors implement an attack scheme called A3CMal using GANs, which creates malware that is capable of being classified as benign by detection schemes. They split their attacks into two groups - targeted and non-targeted. In the targeted attacks, they attempted to force the classifier to label the malware samples with a particular label, while the non-targeted attacks were simply to evade detection, and have the classifier put the malware into a benign category. The existence of an attack such as this, wherein the attackers are able to make the classifier believe the malicious data is something entirely different, chosen from a specific category, is one with serious potential repercussions. Twisting malicious code for a specific classification by an IDS is a very real possibility with the misuse of GANs by malicious actors, and as such is a research problem which requires addressing.

B. WHY NOT GAN?

1) WHEN NOT TO USE GAN

While GAN methods can work extremely well in some situations, there are also some areas and situations in which GAN models will not offer much (if any) improvement. In [141], several open questions into the use of GAN models are posed. One of these is why one would use a GAN model instead of Flow Models, or Autoregressive Models. Odena points out that there are three specific categories for evaluating which of the three to use. This can be seen in the Table 6, which highlights the three metrics proposed by Odena [141].

TABLE 6. Three metrics for determining whether the Generative Adversarial Network model is an appropriate model for a particular task [141].

	Parallel	Efficient	Reversible
GANS	Yes	Yes	No
Flow Models	Yes	No	Yes
Autoregressive Models	No	Yes	Yes

a: TRAINING TRADITIONAL IDS MODELS

When training an Intrusion Detection System, GAN models are of use because they can undertake tasks like generating adversarial examples (see Section VIII-A1), but they are of little to no use in training traditional Intrusion Detection Systems, which do not implement machine learning methods.

b: UNSUITABLE SAMPLES

The suitability of the samples in the dataset used is very important in whether or not to use a GAN model. As in Section VIII-A1, image-based samples are excellent, as are sequences and samples that translate into the image domain without too much computational cost. The most important point here is that if the research involved isn't automatically a suitable data type, the cost of pre-processing that data may be computationally expensive to the point that it is simpler by far to utilize a different type of generative model. Especially when a researcher is looking to create an IDS model which can operate in real-time, the pre-processing requirements for the use of a GAN may simply outweigh the potential gains of employing such a model.

c: ONE SAMPLE, MANY LABELS

While GAN models are excellent at learning contextual clues and semantic relationships, when it comes to output, they are best when there are only a limited number of output "labels". If a sample set has too many potential outcomes, or even has more than one outcome per sample (multilabel classification), GAN models are unlikely to perform well. In these situations it may be more effective and successful to utilize a different generative model. This type of data is less likely to be encountered amongst research into IDS models, but if a researcher is trying to use a GAN on datasets with many different attack classes, rather than merely a Benign/Malicious classification task, the computational power and time requirements may make using a GAN unfeasible.

IX. EMERGING TOPICS

Having discussed when and when not to use GAN models in general research, we now discuss when and where GAN seems to be of most effective use in emerging IDS research. The uses of GAN are many, as seen in Section VII. The most recent areas of development for GANs in Intrusion Detection Systems involve methods for autonomous vehicles (as in [125] among others) and wireless sensor network arrays (such as [119]). These are both critical areas of research with real-world life-or-death outcomes. Sensor networks are deployed throughout Critical National Infrastructure (CNI), and the potential hacking of autonomous vehicles creates the possibility of fatal traffic collisions. In the Russo-Ukrainian War, we have seen the importance of CNI first hand. One scholar argued, in [142], that the employment of cyber-attacks on the CNI of Ukraine by Russia contributed to a "thunder strategy" which helped speed up the war effort. This is an extreme example, which demonstrates the importance of protecting sensor networks and CNI from sophisticated cyber attacks. This is both an area for growth, and an area of great importance, making them an excellent place for researchers to begin exploring ways to utilize the power of GAN models to strengthen CNI against attack.

The employment of GAN models in these areas allows for the adaptation and augmentation of datasets which contain rare classes or which are smaller than may be typical for training neural network models. In newer areas like these, datasets are both rarer and smaller than those for a typical IDS model. As such, the ability to generate more samples becomes an issue of more significance. For one example, in [143] the authors use the Kyoto University Benchmark dataset [144] to train and test their autonomous vehicle IDS. The Kyoto University Benchmark dataset was created in 2006, and contains IDS data taken from traditional computer systems. As such, it is not the ideal dataset for autonomous vehicles, but it is readily available and large enough to train neural network models on. This shows the need for models based on systems like GANs to augment datasets that offer more targeted and vehicle specific samples.

The use of GAN models to create labeled data, as is done in [145], offers a new method of generating large-scale datasets. The requirement for large amounts of labeled data for training and testing of ML models is one of the drawbacks of utilizing these schemes in real-world applications. In a regular scenario, human operators are required to label datasets for use in supervised machine learning. This is both time-intensive and expensive. Thus, the ability to generate labels for existing samples in order to create datasets is a highly important and desirable application of GAN models.

The success in [145] shows the possibilities of GAN for creating realistic data with embedded semantic information. This potential could be transferred to the domain of Intrusion Detection, and offers a potential pathway to new datasets for training and testing. There are also many other avenues for potential research. The methods employed by the authors in [146] to avoid the popular step of translating the dataset into sequences or images and instead working on the data directly using the n-gram feature extraction method is certainly an area worthy of future research for more applications. Any improvements for using GANs without requiring pre-processing data into images offer benefits to domains such as IDS models. While many different methods exist, there is always room for improving the quality and availability of the data, as well as improving the time and computational requirements for processing it.

When it comes to adversarial examples for IDS models, the incredibly low detection rate achieved by [147] shows just how much future research is needed to create IDS models that can successfully fend off attacks from GAN-based systems. Using a GAN attack model can create a situation in which it is possible to test an IDS model against an attacker in real-time, using a controlled environment. This offers plenty of scenarios for improving the performance of IDS models, and especially training them to react appropriately to unseen examples. Working on a pair of ML models as in [91] provides a fully functional scenario in which the researchers can view the full performance of their model.

Overall, the opportunities created by technologies like autonomous vehicles rising to the forefront of public consciousness provide future research directions for those looking at the applications of GAN models in securing network IDS schemes for future technologies. Work done on the vulnerabilities of autonomous vehicles, like that done by Keen Labs (see VII-F) or the examination of vulnerabilities in BMW's more recent autonomous vehicle offerings (see [126]) shows the importance and urgency of research in this area. The prevalence of GAN models for semantic image editing suggests that there is a possibility of utilizing GAN models to edit existing data and perhaps create new attack files using benign traffic. There are significant possibilities for utilizing the high-level semantic information that GANs are capable of capturing in their latent space in order to edit existing data and create new datasets. There are many areas of Network IDS research in GANs that are still developing apace, such as the rapidly expanding world of IoT devices, which offer opportunities for researchers to explore the uses of these machine learning models. Research in Generative Adversarial Networks has exploded in recent years, as researchers have uncovered the many potential applications in numerous fields.

The realms of cybersecurity and intrusion detection contain many possible avenues for research when it comes to GAN algorithms, as has been illustrated in this paper. Our aim is to have provided an explanation of not only what Generative Adversarial Networks are and how they are trained and assessed, but also to have given an effective grounding in the applications within intrusion detection which GANs may work with, both in the current literature and in any potential future research.

X. CONCLUSION

This paper explores the use of Generative Adversarial Networks in research relating to Intrusion Detection Systems, and the potential for optimization therein. We have explored the current models in favor of IDS research; the current research into wired, wireless, mobile, IoT, sensor network, and autonomous vehicle systems; discussed where this research is currently leading; and provided a detailed look at the state-of-the-art as it is in GANs for Network IDS models. This overview of the area explores the ways in which researchers are currently using GANs to improve the performance of these different IDS methods, and the successes and failures they have found through development and exploration. There are several areas of developing research, and many promising methods and implementations. We hope our summation of the current research proves of use to those who are currently in the field of GAN or IDS research, as either a refresher or an introduction to the topic area.

FUNDING

The authors would like to thank the Ministry of Business, Innovation, and Employment (MBIE) from the New Zealand Government to support our work with the grant (MAUX1912) which made it possible for us to conduct the research.

REFERENCES

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–14.

[2] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.

[3] R. Maruzani, "Are you unwittingly helping to train Google's AI models? How Google is using your reCAPTCHA entries to train machine learning models," Medium, 'Towards Data Sci.', Tech. Rep., Jan. 2021.

[4] C. Daly, "'I'm not a robot': Google's anti-robot reCAPTCHA trains their robots to see," AI Bus., Tech. Rep., 2017.

[5] A. Aggarwal, M. Mittal, and G. Battineni, "Generative adversarial network: An overview of theory and applications," *Int. J. Inf. Manage. Data Insights*, vol. 1, no. 1, Apr. 2021, Art. no. 100004.

[6] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F.-Y. Wang, "Generative adversarial networks: Introduction and outlook," *IEEE/CAA J. Autom. Sinica*, vol. 4, no. 4, pp. 588–598, Sep. 2017.

[7] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.

[8] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 16–24, 2013.

[9] S. Smaha, "Haystack: An intrusion detection system," in *Proc. 4th Aerosp. Comput. Secur. Appl.*, Dec. 1988, pp. 37–44.

[10] B. Mukherjee, L. T. Heberlein, and K. N. Levitt, "Network intrusion detection," *IEEE Netw.*, vol. 8, no. 3, pp. 26–41, May 1994.

[11] A. Thakkar and R. Lohiya, "A survey on intrusion detection system: Feature selection, model, performance measures, application perspective, challenges, and future research directions," *Artif. Intell. Rev.*, vol. 55, no. 1, pp. 453–563, Jan. 2022.

[12] M. Alkasassbeh and S. A.-H. Baddar, "Intrusion detection systems: A state-of-the-art taxonomy and survey," *Arabian J. Sci. Eng.*, pp. 1–44, Nov. 2022, doi: 10.1007/s13369-022-07412-1.

[13] A. Arora, "A review on application of GANs in cybersecurity domain," *IETE Tech. Rev.*, vol. 39, no. 2, pp. 433–441, Mar. 2022.

[14] I. K. Dutta, B. Ghosh, A. Carlson, M. Totaro, and M. Bayoumi, "Generative adversarial networks in security: A survey," in *Proc. 11th IEEE Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON)*, Oct. 2020, pp. 0399–0405.

[15] Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li, and Y. Pan, "Generative adversarial networks: A survey toward private and secure applications," *ACM Comput. Surv.*, vol. 54, no. 6, pp. 1–38, Jul. 2022.

[16] S. Baluja and I. Fischer, "Adversarial transformation networks: Learning to generate adversarial examples," 2017, *arXiv:1703.09387*.

[17] Y. Gao and Y. Pan, "Improved detection of adversarial images using deep neural networks," 2020, *arXiv:2007.05573*.

[18] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the GAN: Information leakage from collaborative deep learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 603–618.

[19] Z. Zhao, D. Dua, and S. Singh, "Generating natural adversarial examples," 2017, *arXiv:1710.11342*.

[20] X. Chen, P. Kairouz, and R. Rajagopal, "Understanding compressive adversarial privacy," in *Proc. IEEE Conf. Decis. Control (CDC)*, Dec. 2018, pp. 6824–6831.

[21] C. Huang, P. Kairouz, X. Chen, L. Sankar, and R. Rajagopal, "Context-aware generative adversarial privacy," *Entropy*, vol. 19, no. 12, p. 656, Dec. 2017.

[22] S. Liu, A. Shrivastava, J. Du, and L. Zhong, "Better accuracy with quantified privacy: Representations learned via reconstructive adversarial network," 2019, *arXiv:1901.08730*.

[23] A. Tripathy, Y. Wang, and P. Ishwar, "Privacy-preserving adversarial networks," in *Proc. 57th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2019, pp. 495–505.

[24] K. Alrawashdeh and S. Goldsmith, "Defending deep learning based anomaly detection systems against white-box adversarial examples and backdoor attacks," in *Proc. IEEE Int. Symp. Technol. Soc. (ISTAS)*, Nov. 2020, pp. 294–301.

[25] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," in *Proc. 23rd USENIX Secur. Symp. (USENIX Secur.)*, 2014, pp. 17–32.

[26] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," 2016, *arXiv:1606.03498*.

[27] P. Salehi, A. Chalechale, and M. Taghizadeh, "Generative adversarial networks (GANs): An overview of theoretical model, evaluation metrics, and recent developments," 2020, *arXiv:2005.13178*.

[28] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[29] A. Borji, "Pros and cons of GAN evaluation measures," 2018, *arXiv:1802.03446*.

[30] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li, "Mode regularized generative adversarial networks," 2017, *arXiv:1612.02136*.

[31] *KDD Cup 1999 Data*, Dataset, Canadian Univ. Cybersecur., Univ. New Brunswick, 1999.

[32] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Jul. 2009, pp. 1–6.

[33] *NSL-KDD Dataset*, Canadian Inst. Cybersecur. | Univ. New Brunswick, Fredericton, NB, Canada, 2009.

[34] M. S. Haroon and H. M. Ali, "Adversarial training against adversarial attacks for machine learning-based intrusion detection systems," *Comput., Mater. Continua*, vol. 73, no. 2, pp. 3513–3527, 2022.

[35] D. Stiawan, M. Y. B. Idris, A. M. Bamhdi, and R. Budiarto, "CICIDS-2017 dataset feature analysis with information gain for anomaly detection," *IEEE Access*, vol. 8, pp. 132911–132921, 2020.

[36] S. S. Gopalan, D. Ravikumar, D. Linekar, A. Raza, and M. Hasib, "Balancing approaches towards ML for IDS: A survey for the CSE-CIC IDS dataset," in *Proc. Int. Conf. Commun., Signal Process., Their Appl. (ICCSPA)*, Mar. 2021, pp. 1–6.

[37] *Darpa Intrusion Detection Evaluation Dataset*, Machine Learning in Laboratory, Cambridge, MA, USA, 1999.

[38] R. Robert, E. Marcin, A. Guillem, and S. Thomas, "DARPA dataset | papers with code," Medium, 'Towards Data Sci.', Tech. Rep., Aug. 2022.

[39] M. M. Anjum, S. Iqbal, and B. Hamelin, "Analyzing the usefulness of the DARPA OpTC dataset in cyber threat detection research," in *Proc. 26th ACM Symp. Access Control Models Technol.*, Jun. 2021, pp. 27–32.

[40] O. Yavanoglu and M. Aydos, "A review on cyber security datasets for machine learning algorithms," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2017, pp. 2186–2193.

[41] J. Velasco-Mata, V. González-Castro, E. F. Fernández, and E. Alegre, "Efficient detection of botnet traffic by features selection and decision trees," *IEEE Access*, vol. 9, pp. 120567–120579, 2021.

[42] S. Chowdhury, M. Khanzadeh, R. Akula, F. Zhang, S. Zhang, H. Medal, M. Marufuzzaman, and L. Bian, "Botnet detection using graph-based feature clustering," *J. Big Data*, vol. 4, no. 1, p. 14, Dec. 2017.

[43] A. Bansal and S. Mahapatra, "A comparative analysis of machine learning techniques for botnet detection," in *Proc. 10th Int. Conf. Secur. Inf. Netw.*, Oct. 2017, pp. 91–98.

[44] D. Plohmann, "DGArchive A deep dive into domain generating malware," Fraunhofer FKIE, Tech. Rep., Dec. 2015. [Online]. Available: https://dgarchive.caad.fkie.fraunhofer.de/

[45] C. Choudhary, R. Sivaguru, M. Pereira, B. Yu, A. C. Nascimento, and M. De Cock, "Algorithmically generated domain detection and malware family classification," in *Security in Computing and Communications: 6th International Symposium, SSCC 2018, Bangalore, India, September 19–22, 2018, Revised Selected Papers 6*. Singapore: Springer, 2019, pp. 640–655.

[46] A. O. Almashhadani, M. Kaiiali, D. Carlin, and S. Sezer, "MaldomDetector: A system for detecting algorithmically generated domain names with machine learning," *Comput. Secur.*, vol. 93, Jun. 2020, Art. no. 101787.

[47] R. Mutalik, D. Chheda, Z. Shaikh, and D. Toradmalle, "RockYou," Dataset, SkullSecur., Rapid7, 2010.

[48] B. Hitaj, P. Gasti, G. Ateniese, and F. Perez-Cruz, "PassGAN: A deep learning approach for password guessing," 2017, *arXiv:1709.00440*.

[49] D. Biesner, K. Cvejoski, B. Georgiev, R. Sifa, and E. Krupicka, "Generative deep learning techniques for password generation," 2020, *arXiv:2012.05685*.

[50] G. Creech and J. Hu, "Generation of a new IDS test dataset: Time to retire the KDD collection," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2013, pp. 4487–4492.

[51] G. Creech and J. Hu, "A semantic approach to host-based intrusion detection systems using contiguousand discontiguous system call patterns," *IEEE Trans. Comput.*, vol. 63, no. 4, pp. 807–819, Apr. 2014.

[52] G. Creech, "Developing a high-accuracy cross platform host-based intrusion detection system capable of reliably detecting zero-day attacks," Ph.D. thesis, School Eng. Inf. Technol., UNSW Sydney, Sydney, NSW, Australia, 2014.

[53] T. Mouttaqi, T. Rachidi, and N. Assem, "Re-evaluation of combined Markov-bayes models for host intrusion detection on the ADFA dataset," in *Proc. Intell. Syst. Conf. (IntelliSys)*, Sep. 2017, pp. 1044–1052.

[54] R. A. Khamis and A. Matrawy, "Evaluation of adversarial training on different types of neural networks in deep learning-based IDSs," in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, Oct. 2020, pp. 1–6.

[55] Z. Zoghi and G. Serpen, "UNSW-NB15 computer security dataset: Analysis through visualization," 2021, *arXiv:2101.05067*.

[56] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, Nov. 2015, pp. 1–6.

[57] R. Durall, A. Chatzimichailidis, P. Labus, and J. Keuper, "Combating mode collapse in GAN training: An empirical analysis using Hessian eigenvalues," 2020, *arXiv:2012.09673*.

[58] H. Thanh-Tung and T. Tran, "Catastrophic forgetting and mode collapse in GANs," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–10.

[59] A. Seff, A. Beatson, D. Suo, and H. Liu, "Continual learning in generative adversarial nets," 2017, *arXiv:1705.08395*.

[60] J. Gauthier, "Conditional generative adversarial nets for convolutional face generation," in *Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter Semester*, vol. 2014, no. 5. San Francisco, CA, USA: Stanford Univ., 2014, p. 2.

[61] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*.

[62] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," 2014, *arXiv:1412.6806*.

[63] A. Mordvintsev, C. Olah, and M. Tyka, "Inceptionism: Going deeper into neural networks," Google Res. Labs, Tech. Rep., 2015.

[64] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[65] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 214–223.

[66] R. Chauhan, U. Sabeel, A. Izaddoost, and S. S. Heydari, "Polymorphic adversarial cyberattacks using WGAN," *J. Cybersecurity Privacy*, vol. 1, no. 4, pp. 767–792, Dec. 2021.

[67] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," in *Proc. 5th Int. Conf. Learn. Represent.*, 2017, pp. 1–18.

[68] Q. Yang and X. Li, "BiGAN: LncRNA-disease association prediction based on bidirectional generative adversarial network," *BMC Bioinf.*, vol. 22, no. 1, p. 357, Dec. 2021.

[69] W. Xu, J. Jang-Jaccard, T. Liu, and F. Sabrina, "Training a bidirectional GAN-based one-class classifier for network intrusion detection," 2022, *arXiv:2202.01332*.

[70] G. Renjith, S. Laudanna, S. Aji, C. Visaggio, and P. Vinod, "GANG-MAM: GAN based enGine for modifying Android malware," *SoftwareX*, vol. 18, Jun. 2022, Art. no. 100977.

[71] H. Rafiq, N. Aslam, B. Issac, and R. H. Randhawa, "An investigation on fragility of machine learning classifiers in Android malware detection," in *Proc. IEEE INFOCOM Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, May 2022, pp. 1–6.

[72] M. M. Zhu, S. Gong, Z. Qian, and L. Zhang, "A brief review on cycle generative adversarial networks," in *Proc. 7th Int. Conf. Intell. Syst. Image Process.*, 2019, pp. 235–242.

[73] Y. Chen, Y. Pan, T. Yao, X. Tian, and T. Mei, "Mocycle-GAN: Unpaired video-to-video translation," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 647–655.

[74] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," 2016, *arXiv:1610.09585*.

[75] R. Nagaraju and M. Stamp, "Auxiliary-classifier GAN for malware analysis," 2021, *arXiv:2107.01620*.

[76] S. Kausar, B. Tahir, and M. A. Mehmood, "HashCat: A novel approach for the topic classification of multilingual Twitter trends," in *Proc. Int. Conf. Frontiers Inf. Technol. (FIT)*, Dec. 2021, pp. 212–217.

[77] R. Hranický, L. Zobal, O. Ryšavý, and D. Kolář, "Distributed password cracking with BOINC and hashcat," *Digit. Invest.*, vol. 30, pp. 161–172, Sep. 2019.

[78] L. Yan, W. Zheng, C. Gou, and F.-Y. Wang, "IsGAN: Identity-sensitive generative adversarial network for face photo-sketch synthesis," *Pattern Recognit.*, vol. 119, Nov. 2021, Art. no. 108077.

[79] D. Berthelot, T. Schumm, and L. Metz, "BEGAN: Boundary equilibrium generative adversarial networks," 2017, *arXiv:1703.10717*.

[80] H. Gao, J. Pei, and H. Huang, "ProGAN: Network embedding via proximity generative adversarial network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 1308–1316.

[81] A. Karnewar and O. Wang, "MSG-GAN: Multi-scale gradients for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7796–7805.

[82] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," 2018, *arXiv:1805.08318*.

[83] Y. Chen, Q. Gao, and X. Wang, "Inferential Wasserstein generative adversarial networks," *J. Roy. Stat. Soc. Ser. B, Stat. Methodol.*, vol. 84, no. 1, pp. 83–113, Feb. 2022.

[84] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," 2016, *arXiv:1606.03657*.

[85] L. Yu, W. Zhang, J. Wang, and Y. Yu, "SeqGAN: Sequence generative adversarial nets with policy gradient," 2016, *arXiv:1609.05473*.

[86] Y. Chen, Y. Xiong, B. Liu, and X. Yin, "TranGAN: Generative adversarial network based transfer learning for social tie prediction," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[87] M. Garuba, C. Liu, and D. Fraites, "Intrusion techniques: Comparative study of network intrusion detection systems," in *Proc. 5th Int. Conf. Inf. Technol., New Generat. (itng)*, Apr. 2008, pp. 592–598.

[88] W. Xu, J. Jang-Jaccard, T. Liu, F. Sabrina, and J. Kwak, "Improved bidirectional GAN-based approach for network intrusion detection using one-class classifier," *Computers*, vol. 11, no. 6, p. 85, May 2022.

[89] J. Yang, T. Li, G. Liang, W. He, and Y. Zhao, "A simple recurrent unit model based intrusion detection system with DCGAN," *IEEE Access*, vol. 7, pp. 83286–83296, 2019.

[90] S. P. Kulyadi, P. Mohandas, S. K. S. Kumar, M. J. S. Raman, and V. S. Vasan, "Anomaly detection using generative adversarial networks on firewall log message data," in *Proc. 13th Int. Conf. Electron., Comput. Artif. Intell. (ECAI)*, Jul. 2021, pp. 1–6.

[91] M. Usama, M. Asim, S. Latif, and J. Qadir, "Generative adversarial networks for launching and thwarting adversarial attacks on network intrusion detection systems," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2019, pp. 78–83.

[92] C. Choi, S. Shin, and I. Lee, "Opcode sequence amplifier using sequence generative adversarial networks," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2019, pp. 968–970.

[93] Y. Liu, J. Li, B. Liu, X. Gao, and X. Liu, "Malware identification method based on image analysis," in *Proc. 11th Int. Conf. Inf. Technol. Med. Educ. (ITME)*, Nov. 2021, pp. 157–161.

[94] S. Wang, Q. Wang, Z. Jiang, X. Wang, and R. Jing, "A weak coupling of semi-supervised learning with generative adversarial networks for malware classification," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 3775–3782.

[95] C. Forensics, "VirusShare–because sharing is caring," Database Repository, Corvus Forensics, New York, NY, USA, Tech. Rep.

[96] R. Ronen, M. Radu, C. Feuerstein, E. Yom-Tov, and M. Ahmadi, "Microsoft malware classification challenge (BIG 2015)," Feb. 2018, *arXiv:1802.10135*.

[97] X. Peng, H. Xian, Q. Lu, and X. Lu, "Semantics aware adversarial malware examples generation for black-box attacks," *Appl. Soft Comput.*, vol. 109, Sep. 2021, Art. no. 107506.

[98] V. S. Bhaskara and D. Bhattacharyya, "Emulating malware authors for proactive protection using GANs over a distributed image visualization of dynamic file behavior," 2018, *arXiv:1807.07525*.

[99] W. L. Tan and T. Truong-Huu, "Enhancing robustness of malware detection using synthetically-adversarial samples," in *Proc. GLOBECOM IEEE Global Commun. Conf.*, Dec. 2020, pp. 1–6.

[100] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville, "Adversarially learned inference," 2016, *arXiv:1606.00704*.

[101] J.-T. Wang and C.-H. Wang, "High performance WGAN-GP based multiple-category network anomaly classification system," in *Proc. Int. Conf. Cyber Secur. Emerg. Technol. (CSET)*, Oct. 2019, pp. 1–7.

[102] Q. Yan, M. Wang, W. Huang, X. Luo, and F. R. Yu, "Automatically synthesizing DoS attack traces using generative adversarial networks," *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 12, pp. 3387–3396, Dec. 2019.

[103] C. Park, J. Lee, Y. Kim, J.-G. Park, H. Kim, and D. Hong, "An enhanced AI-based network intrusion detection system using generative adversarial networks," *IEEE Internet Things J.*, vol. 10, no. 3, pp. 2330–2345, Feb. 2023.

[104] J. Lee and K. Park, "GAN-based imbalanced data intrusion detection system," *Pers. Ubiquitous Comput.*, vol. 25, no. 1, pp. 121–128, Feb. 2021.

[105] Z. Wang, P. Wang, X. Zhou, S. Li, and M. Zhang, "FLOWGAN: Unbalanced network encrypted traffic identification method based on GAN," in *Proc. IEEE Int. Conf Parallel Distrib. Process. With Appl., Big Data Cloud Comput., Sustain. Comput. Commun., Social Comput. Netw. (ISPA/BDCloud/SocialCom/SustainCom)*, Dec. 2019, pp. 975–983.

[106] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of encrypted and VPN traffic using time-related features," in *Proc. 2nd Int. Conf. Inf. Syst. Secur. Privacy*, 2016, pp. 407–414.

[107] J. Lee and K. Park, "AE-CGAN model based high performance network intrusion detection system," *Appl. Sci.*, vol. 9, no. 20, p. 4221, Oct. 2019.

[108] S. Zhao, J. Li, J. Wang, Z. Zhang, L. Zhu, and Y. Zhang, "AttackGAN: Adversarial attack against black-box IDS using generative adversarial networks," *Proc. Comput. Sci.*, vol. 187, pp. 128–133, Jan. 2021.

[109] A. Ferdowsi and W. Saad, "Generative adversarial networks for distributed intrusion detection in the Internet of Things," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[110] N. Abdalgawad, A. Sajun, Y. Kaddoura, I. A. Zualkernan, and F. Aloul, "Generative deep learning to detect cyberattacks for the IoT-23 dataset," *IEEE Access*, vol. 10, pp. 6430–6441, 2022.

[111] S. Garcia, A. Parmisano, and M. J. Erquiaga, *IoT-23: A Labeled Dataset With Malicious and Benign IoT Network Traffic*. Honolulu, HI, USA: Zenodo, 2020.

[112] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, and M. Kallitsis, "Understanding the Mirai botnet," in *Proc. 26th {USENIX} Secur. Symp. ({USENIX} Secur.)*, 2017, pp. 1093–1110.

[113] T. Zixu, K. S. K. Liyanage, and M. Gurusamy, "Generative adversarial network and auto encoder based anomaly detection in distributed IoT networks," in *Proc. GLOBECOM IEEE Global Commun. Conf.*, Dec. 2020, pp. 1–7.

[114] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," 2018, *arXiv:1811.00701*.

[115] N. Leavitt, "Mobile security: Finally a serious problem?" *Computer*, vol. 44, no. 6, pp. 11–14, Jun. 2011.

[116] S. Wei, P. Jiang, Q. Yuan, and J. Wang, "Mobile application network behavior detection and evaluation with WGAN and bi-LSTM," in *Proc. TENCON IEEE Region Conf.*, Oct. 2018, pp. 44–49.

[117] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," 2018, *arXiv:1803.04311*.

[118] C. Zhang, X. Ouyang, and P. Patras, "ZipNet-GAN: Inferring fine-grained mobile traffic patterns via a generative adversarial neural network," in *Proc. 13th Int. Conf. Emerg. Netw. Exp. Technol.*, Nov. 2017, pp. 363–375.

[119] S. M. W. Umba, A. M. Abu-Mahfouz, T. D. Ramotsoela, and G. P. Hancke, "A review of artificial intelligence based intrusion detection for software-defined wireless sensor networks," in *Proc. IEEE 28th Int. Symp. Ind. Electron. (ISIE)*, Jun. 2019, pp. 1277–1282.

[120] P. R. Grammatikis, P. Sarigiannidis, G. Efstathopoulos, and E. Panaousis, "ARIES: A novel multivariate intrusion detection system for smart grid," *Sensors*, vol. 20, no. 18, p. 5305, Sep. 2020.

[121] M. Rudner, "Cyber-threats to critical national infrastructure: An intelligence challenge," *Int. J. Intell. CounterIntell.*, vol. 26, no. 3, pp. 453–481, Sep. 2013.

[122] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy*, 2018, pp. 108–116.

[123] A. Greenberg, "Hackers remotely kill a Jeep on the highway—With me in it," WIRED, Jul. 21, 2015.

[124] D. Shepardson, "Fiat chrysler will recall vehicles over hacking worries," Reuters, 2015.

[125] K. Kim, J. S. Kim, S. Jeong, J.-H. Park, and H. K. Kim, "Cybersecurity for autonomous vehicles: Review of attacks and defense," *Comput. Secur.*, vol. 103, Apr. 2021, Art. no. 102150.

[126] Z. Cai, A. Wang, W. Zhang, M. Gruffke, and H. Schweppe, "0-Days & mitigations: Roadways to exploit and secure connected BMW cars," *Black Hat USA*, vol. 2019, p. 39, Aug. 2019.

[127] E. Seo, H. M. Song, and H. K. Kim, "GIDS: GAN based intrusion detection system for in-vehicle network," in *Proc. 16th Annu. Conf. Privacy, Secur. Trust (PST)*, Aug. 2018, pp. 1–6.

[128] H. Sedjelmaci, "Attacks detection and decision framework based on generative adversarial network approach: Case of vehicular edge computing network," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 10, Oct. 2022, Art. no. e4073.

[129] C. Smutz and A. Stavrou, "Malicious PDF detection using metadata and structural features," in *Proc. 28th Annu. Comput. Secur. Appl. Conf.*, Dec. 2012, pp. 239–248.

[130] H. Bae, Y. Lee, Y. Kim, U. Hwang, S. Yoon, and Y. Paek, "Learn2Evade: Learning-based generative model for evading PDF malware classifiers," *IEEE Trans. Artif. Intell.*, vol. 2, no. 4, pp. 299–313, Aug. 2021.

[131] X. Zhang, J. Wang, M. Sun, and Y. Feng, "AndrOpGAN: An opcode GAN for Android malware obfuscations," in *Proc. Int. Conf. Mach. Learn. Cyber Secur.*, vol. 12486, 2020, pp. 12–25.

[132] W. Hu and Y. Tan, "Generating adversarial malware examples for black-box attacks based on GAN," 2017, *arXiv:1702.05983*.

[133] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 53–65, Jan. 2018.

[134] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," 2017, *arXiv:1711.09020*.

[135] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas, "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks," 2016, *arXiv:1612.03242*.

[136] A. Cherepkov, A. Voynov, and A. Babenko, "Navigating the GAN parameter space for semantic image editing," 2020, *arXiv:2011.13786*.

[137] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 12, pp. 4217–4228, Dec. 2021.

[138] M. Amin, B. Shah, A. Sharif, T. Ali, K.-I. Kim, and S. Anwar, "Android malware detection through generative adversarial networks," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 2, Feb. 2022, Art. no. e3675.

[139] R. Chauhan and S. S. Heydari, "Polymorphic adversarial DDoS attack on IDS using GAN," in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, Oct. 2020, pp. 1–6.

[140] Z. Fang, J. Wang, J. Geng, Y. Zhou, and X. Kan, "A3CMal: Generating adversarial samples to force targeted misclassification by reinforcement learning," *Appl. Soft Comput.*, vol. 109, Sep. 2021, Art. no. 107505.

[141] A. Odena, "Open questions about generative adversarial networks," *Distill*, vol. 4, no. 4, p. e18, Apr. 2019.

[142] G. Mueller, B. Jensen, B. Valeriano, R. Maness, and J. Macias, "Cyber operations during the Russo–Ukrainian war," Center for Strategic Int. Studies, Washington, DC, USA, 2023.

[143] K. M. A. Alheeti and K. McDonald-Maier, "Intelligent intrusion detection in external communication systems for autonomous vehicles," *Syst. Sci. Control Eng.*, vol. 6, no. 1, pp. 48–56, Jan. 2018.

[144] J. Song, H. Takakura, and Y. Okabe, "Description of Kyoto University benchmark data," 2006. [Online]. Available: http://www.takakura.com/Kyoto_data/BenchmarkData-Description-v5.pdf

[145] L. Sixt, B. Wild, and T. Landgraf, "RenderGAN: Generating realistic labeled data," *Frontiers Robot. AI*, vol. 5, p. 66, Jun. 2018.

[146] E. Zhu, J. Zhang, J. Yan, K. Chen, and C. Gao, "N-gram MalGAN: Evading machine learning detection via feature n-gram," *Digit. Commun. Netw.*, vol. 8, no. 4, pp. 485–491, Aug. 2022.

[147] X. Li, K. Kong, S. Xu, P. Qin, and D. He, "Feature selection-based Android malware adversarial sample generation and detection method," *IET Inf. Secur.*, vol. 15, no. 6, pp. 401–416, Nov. 2021.

AERYN DUNMORE (Graduate Student Member, IEEE) received the master's degree in computing and information sciences from the Auckland University of Technology, in 2017. She is currently pursuing the Ph.D. degree. Her Ph.D. dissertation was on creating alternative encryption systems, titled "Using Graphic Based Systems to Improve Cryptographic Algorithms." She is a Research Assistant with Massey University. She specializes in neural networks for cybersecurity and encryption designs. She has studied at the University of Auckland and Oxford University.

JULIAN JANG-JACCARD received the M.Sc. and Ph.D. degrees from the University of Sydney, Australia. She is currently an Associate Professor and the Head of the Cybersecurity Laboratory at Massey University, New Zealand. She has published more than 70 papers in leading conferences and journal venues, including IEEE and ACM. Her research interests include cybersecurity, intrusion detection, anomaly detection, artificial intelligence, data anonymization, and privacy-preservation techniques. She was a recipient of many multi-million dollar research awards both from Australian and New Zealand governments/industries while collaborating with the top international ICT companies and universities around the world.

FARIZA SABRINA (Member, IEEE) received the M.E. degree (by research) in electrical and information engineering from the University of Sydney, Australia, and the Ph.D. degree in computer science and engineering from the University of New South Wales, Australia. She has many years of research, teaching, and industrial experience in information and communication technologies. She is currently a Senior Lecturer and the Discipline Lead of Network and Information Security with the School of Engineering and Technology, Central Queensland University, Australia. Her current research interests include networking and information security, the Internet of Things (IoT), cybersecurity, blockchain, and artificial intelligence. She serves as a technical program committee member of various conferences. She is a member of ACM and ACS.

JIN KWAK is a Professor and the Head of the Department of Cybersecurity at Ajou University, Republic of Korea. He has more than 150 publications in leading journals and conferences. His current research interests include authentication, information security and privacy, applied cryptography, wireless security, and data encryption.