electronics

MDPI

Article

A Survey on Data-Driven Learning for Intelligent Network Intrusion Detection Systems

Ghada Abdelmoumin [1,*,†,‡], Jessica Whitaker [1,†,‡], Danda B. Rawat [1,†,‡] and Abdul Rahman [2,‡]

[1] Data Science and Cybersecurity Center, Department of Electrical Engineering and Computer Science, Howard University, Washington, DC 20059, USA; jessica.whitaker1@bison.howard.edu (J.W.); danda.rawat@howard.edu (D.B.R.)

[2] Microsoft Corporation, Reston, VA 20190, USA; abdulrahman@microsoft.com

* Correspondence: ghada.abdelmoumin@bison.howard.edu

† Current address: Department of Electrical Engineering and Computer Science, School of Engineering and Architecture, Howard University, 2300 Sixth Street NW, Washington, DC 20059, USA.

‡ These authors contributed equally to this work.

Abstract: An effective anomaly-based intelligent IDS (AN-Intel-IDS) must detect both known and unknown attacks. Hence, there is a need to train AN-Intel-IDS using dynamically generated, real-time data in an adversarial setting. Unfortunately, the public datasets available to train AN-Intel-IDS are ineluctably static, unrealistic, and prone to obsolescence. Further, the need to protect private data and conceal sensitive data features has limited data sharing, thus encouraging the use of synthetic data for training predictive and intrusion detection models. However, synthetic data can be unrealistic and potentially bias. On the other hand, real-time data are realistic and current; however, it is inherently imbalanced due to the uneven distribution of anomalous and non-anomalous examples. In general, non-anomalous or normal examples are more frequent than anomalous or attack examples, thus leading to skewed distribution. While imbalanced data are commonly predominant in intrusion detection applications, it can lead to inaccurate predictions and degraded performance. Furthermore, the lack of real-time data produces potentially biased models that are less effective in predicting unknown attacks. Therefore, training AN-Intel-IDS using imbalanced and adversarial learning is instrumental to their efficacy and high performance. This paper investigates imbalanced learning and adversarial learning for training AN-Intel-IDS using a qualitative study. It surveys and synthesizes generative-based data augmentation techniques for addressing the uneven data distribution and generative-based adversarial techniques for generating synthetic yet realistic data in an adversarial setting using rapid review, structured reporting, and subgroup analysis.

Keywords: imbalanced learning; adversarial learning; generative models; generative adversarial networks; oversampling; intrusion detection systems; machine learning; deep learning

check for updates

Citation: Abdelmoumin, G.; Whitaker, J.; Rawat, D.B.; Rahman, A. A Survey on Data-Driven Learning for Intelligent Network Intrusion Detection Systems. *Electronics* **2022**, *11*, 213. https://doi.org/10.3390/electronics11020213

Academic Editor: Amir Mosavi

Received: 7 December 2021
Accepted: 6 January 2022
Published: 11 January 2022

1. Introduction

In a binary classification problem, such as anomaly-based detection, where the dataset contains two sets of examples (normal and anomalous), it is common to encounter class imbalance. Class imbalance generally occurs when the normal set contains significantly more examples or samples than the anomalous set, thus dividing the dataset into minority and majority class samples. While both classes exist in binary classification datasets, the minority class is the one that is often of interest to many binary classification problems [1]. For example, training an AN-Intel-IDS model often involves imbalanced data where the normal samples are typically more frequent than the anomalous. Even when considering realistic conditions, the normal class contains numerous samples compared to the abnormal or anomalous class, which is often poorly sampled or not well-defined [2]. Inevitably, the training dataset influences the machine learning models' predictability and performance. Hence, an imbalanced dataset can lead to classification issues, such as over classification. For example, data imbalance or uneven class distribution can cause an AN-Intel-IDS model to over classify the normal class due to its high probability in the dataset compared to the anomalous one. Thus, data imbalance directly impacts the trained model's prediction accuracy and overall performance. Further, the class imbalance has an adverse impact on many machine learning (ML) algorithms [3]. It can lead to biased predictive models, misclassification, and performance degradation. For instance, an anomaly-based detection model is usually biased towards the majority class (i.e., the normal or non-malicious class). In addition to data imbalance, training ML models using insufficient data can also lead to misclassification and performance degradation issues. In general, the expensiveness of data collection and gathering often creates a situation where data for training learning models becomes insufficiently large to train these models effectively. Invariably, insufficient data and imbalanced data adversely affect ML and deep learning (DL) models, where the models' performance typically degrades when learning from severely imbalanced data, insufficient data, or both [3–9]. Therefore, training effective AN-Intel-IDS models require considerable data [3]. However, sufficient data for training learning models may not be readily available due to data sparsity, data limitation, data privacy, and data sensitivity. Invariably, data privacy, concealment, and sensitivity limit data sharing and accessibility, leading to the use of synthetic data to train the predictive models instead of actual or ground truth data.

Using synthetic data to train, test, or evaluate AN-Intel-IDS models is problematic when the training of predictive models typically involves imbalanced data. In addition, synthetic data can be unrealistic and potentially bias. On the other hand, ground-truth datasets are highly desirable when training AN-Intel-IDS models. However, the ground truth data usually has a skewed distribution, thus, leading to biased predictive models. Another critical issue to consider when training AN-Intel-IDS models is the ability of the learning model to detect both known and unknown attacks, which indicates the need to train learning models in an adversarial setting by generating novel adversarial examples containing unforeseen attacks. Hence, adversarial learning is as equally crucial as imbalanced learning. Nonetheless, adversarial learning still suffers from data imbalance. In general, anomaly-based detection models use a one-class approach, typically the normal class, to detect intrusions even when the abnormal class is of interest. Further, even when using adversarial learning approaches, such as unsupervised generative adversarial networks (GAN), the underlying assumption is that the training data are anomaly-free [4]. As a result, anomaly-based models suffer from false positives due to the model's bias towards the normal examples.

Traditional approaches to solving the data imbalance problem had primarily focused on data augmentation, data generation, and data imputation. Data augmentation aims at increasing the data by creating additional training samples either by transforming the data in the data space or creating additional examples in the feature space [5]. Data generation aims to increase the data by creating new synthetic data to preserve the privacy and confidentiality of the actual data. Data imputation, which targets missing data, increases data samples by replacing missing values using substitution methods, such as regression and matching methods, to name a few [6,7]. While data augmentation, data generation, and data imputation are the intuitive approaches to consider, generating sufficient data or obtaining a balanced class distribution from ground truth data using traditional ML or DL is invariably tortuous. Hence, a better approach is to use non-traditional ML and DL methods to address data imbalance for normal learning and adversarial learning. For example, non-traditional ML and DL methods for data augmentation, data generation, or both include but not limited to IoT Sequential GAN, Wasserstein GAN plus Gradient Penalty (WGAN-GP), and GAN plus Synthetic Minority Oversampling Techniques (GAN-SMOTE) [10–12].

This paper surveys the most recent data augmentation and generation methods, we refer to as data-driven learning (DDL) methods, for imbalanced and adversarial learning using rapid review, structured reporting, and subgroup analysis. It focuses on methods that employ non-traditional approaches to data augmentation and generation, such as generative models, and whose domain of application is one or more of the following domains: Internet of Things (IoT), cyber-physical systems (CPS), smart homes, intrusion detection systems, traditional communication networks, and cybersecurity. In general, this paper aims to present a rapid review of data augmentation and generation methods proposed, particularly in the last three to four years using a qualitative approach. Its main goal is to enable researchers, who are interested in using data-driven learning to address data scarcity and data restriction present to build AN-Intel IDS, to get a general idea of the latest proposed methods using non-traditional ML and DL approaches, as well as a general understanding of the challenges and knowledge of open issues. Hence, the main contribution of this paper is a classification scheme based on the method class of learning, type of the generative model, the specific domain of application, publication year, and subgroup analysis. Subgroup analysis uses the type of approach to subgroup the methods, to show the evaluation results based on the following: data quality, prediction accuracy using f-score, and performance compared to other methods. In addition, it highlights the advantages and disadvantages of DDL methods. The remainder of this paper's organization is as follows: In Section 2, we provide background preliminaries. Then, we review the related work in Section 3 and describe our research design and methodology in Section 4. Next, in Sections 5–7, we describe and summarize the DDL methods for imbalanced and adversarial learning, respectively. Then, we present the results of our findings in Section 8. Finally, we highlight challenges, discuss open research issues in Section 9 and conclude in Section 10.

2. Background Preliminaries

Data resampling techniques help solve the problem of imbalanced data. Whereas data augmentation techniques help solve data imbalance, data scarcity and enable adversarial training. In this section, we describe resampling and augmentation techniques that focus on preliminaries limited to the surveyed methods. Additionally, we focus on techniques that were widely mentioned and used in the reviewed literature.

2.1. Data Resampling Techniques

Resampling techniques, which are often applicable before learning, adjust the minority class distribution to solve the data imbalance problem. Examples of data resampling techniques include random oversampling, random undersampling, and SMOTE. Oversampling and undersampling techniques focus on balancing the distribution of the majority and minority classes in the dataset. Oversampling increases the weight of the minority class, whereas undersampling reduces the weight of the majority class [8]. While oversampling and undersampling reduce data imbalance using the same dataset, SMOTE, an intelligent data resampling technique, reduces the degree of imbalance by synthetically creating a new minority class [13]. The authors in [8] define SMOTE as an oversampling technique, whereas the authors in [13] define SMOTE as a combination of data oversampling and data undersampling techniques. Each one of the three resampling techniques suffers from a specific issue. Oversampling increases the data size and can lead to learner model overfitting. Undersampling reduces the data size; however, it leads to information loss. SMOTE suffers from overfitting and overlapping. However, overfitting is less significant in SMOTE than overlapping, which results from interpolating between relatively adjacent instances of the minority class.

2.2. Data Augmentation Techniques

Data augmentation techniques can solve data imbalance without suffering from overlapping or resulting in model overfitting. For example, techniques such as GAN address oversampling and overfitting issues by specifying the resampling rate ahead of time and using noise to increase the minority class examples. Given this prior knowledge, GANS can replicate the data [8,10]. Apart from GAN, this paper considers other data augmentation techniques such as autoencoders (AE) and Wesserian GAN (WGAN). In doing so, we focus on key preliminaries limited to the surveyed methods.

GAN, which belongs to the class of generative models, is a priori knowledge method that uses neural networks to generate data from noise. Its main goal is to learn the data distribution and then mimic the distribution to either create a similar distribution or a variant of it [14]. Historically, the use of GANs focused on generating adversarial examples (normal examples with added perturbations) to deceive an image classifier. However, its use extends beyond adversarial learning and the image recognition domain. GANs can generate benign data, adversarial data, or both, and their application can span other domains such as networking and cybersecurity. A basic GAN consists of a generator and discriminator; both are neural networks locked in a min-max game. The generator tries to maximize the loss function, and the discriminator minimizes it. In this min-max game, the generator goal is to generate plausible data indistinguishable from authentic data. The discriminator goal is to classify the generated data as plausible or implausible [15].

A variation of a GAN, known as Wasserstein GAN (WGAN), can model discrete distributions over a continuous feature space [11]. WGAN trains the discriminator to discriminate between plausible and implausible examples using an estimation rather than a classification, i.e., outputting a number, which is significantly large when data are authentic and small otherwise. Similar to GANs, Autoencoders adds a small perturbation to the input. They are generative models that consist of two neural networks known as an encoder and a decoder [16]. The encoder learns how to add noise to efficiently encode the data, while the decoder learns how to decode the encoded data by distinguishing between added noise and original data. If the encoded data are different from the decoded data, the autoencoder adjusts its weight. Autoencoders have performance comparable to GANs and have their use in anomaly detection.

In general, GANs can address data imbalance by generating more data without exhibiting overlapping or overfitting. Unlike the resampling and SMOTE techniques, a GAN can solve the data imbalance by specifying the sampling rate and replicating existing data rather than generating new data with a feature space closer to the existing one without specifying the sampling rate.

3. Related Work

The thesis in [17], which focused on network security for IoT, provided an overview of generative deep learning models for generating network traffic. The author broadly categorized the surveyed models into network flow-level and network packet-level. Further, they classified the models based on the type of the generated network traffic, the employed algorithm, and used features of the generated traffic. In addition, the author highlighted the limitations of these models based on IoT traffic generation and the level of generated traffic and proposed a hybrid model that generated a combination of flow-level and packet-level network traffic.

The authors in [18] focused on GAN-based anomaly detection (AnoGAN) methods by highlighting their pros and cons. The authors suggested that the GAN-based methods for anomaly detection's approach built on the adversarial feature learning approach for detecting anomaly used by the bidirectional GAN, known as BiGAN. Further, they empirically validated the main GAN-based models for anomaly detection by re-implementing all models and evaluating their performance using the commonly known datasets for training and testing the models.

The study in [19] focused on adversarial examples for deceiving deep learning models for image recognition and surveyed AEs generation methods and their defense techniques. The authors suggested that one notable aspect of AEs was that the same set of AEs could attack different models with different architecture and training data. Further, they explained the cause of AEs, described their characteristics, and discussed their evaluation metrics. Additionally, the authors listed AEs' adversarial abilities and goals and introduced AEs' construction methods, highlighting their advantages and disadvantages. The authors compared their attributes, success rate, and transfer rate based on different attack methods. Moreover, they described the primary goals of defending against AEs, detailed current defense techniques and their limitations, and summarized several challenges.

Focusing on the creation of synthetic data through deep generative models, the authors in [3] provided a comprehensive survey of GAN-based approaches for generating or transforming synthetic network data for network applications such as IoT and mobile networks and presented an overall taxonomy of generative models where they broadly divided them into explicit density and implicit density models. Further, they provided a detailed overview of GAN variations and architectures and their applications in computer and communication networks. They proposed an evaluation framework for comparing the performance of different GAN-based approaches using publicly known network datasets. Most notably, they provided a taxonomic categorization of generative approaches based on their application, problem solved, and model used over the various classes of mobile network, network analysis, IoT, physical layer, and cybersecurity, In addition, they introduced parameters for evaluating GANs such as loss, optimizer, learning rate, latent dimension, batch size, and epochs.

The paper in [20] reviewed GANs and discussed their strength compared to other generative models and how they operate. The authors noted problems related to the training, testing, and evaluation of GANs and further classified the GANs based on the used approach into two categories: GANs to protect cybersecurity systems from attacks and GANs used to attack cybersecurity systems. In addition, they highlighted four GAN properties and discussed variant GAN architectures and GANs limitations in cybersecurity applications.

The study in [21] focused on the field of imbalanced learning development by discussing data imbalance open issues and challenges related to various forms of learning and new methods for managing data imbalance for recent applications. The author analyzed different aspects of imbalanced learning such as classification, clustering, and regression and highlighted challenges in several critical areas, including imbalanced classification and semi-supervised and unsupervised handling of imbalanced datasets.

The authors in [13] focused on high-class imbalance, where the majority to minority class ratio is 100:1 and 10,000:1, in big data. Further, they discussed data-level and algorithm-level techniques and reviewed methods addressing the class imbalance in regular and big data. The authors noted that data sampling methods with random over-sampling methods showed overall better results concerning the class imbalance. However, algorithm-level methods performance reported in the literature showed inconsistent and conflicting results and evaluation methods with limited scope. As a result, the authors suggested the need for comprehensive, comparative studies.

This study focuses on data scarcity, unequal data distribution, lack of adversarial examples and survey generative approaches, data generation, data augmentation, imputation methods for training, testing, and validating intelligent network intrusion detection systems using non-adversarial and adversarial settings. While it expands on the scope of DDL methods, it is not exhaustive and does not comparatively analyze and evaluate the performance of the surveyed methods. However, it provides a subgroup analysis, where the type of the approach indicates the method's subgroup membership, to enable comparing studies based on their evaluation results, such as standard measures to assess the augmented or generated data's quality, and the trained models' prediction accuracy and performance compared to other methods, algorithms, or approaches.

4. Research Design and Methodology

This section describes the design method and research methodology we used conducting the survey, extracting the existing research work on DDL methods for imbalanced and adversarial learning, synthesizing information, and reporting findings.

For our methodological approach, we used a rapid review method to survey the literature on non-traditional or generative-based data augmentation and generation methods for imbalanced and adversarial learning. Precluding meta-analysis, we used structured reporting and subgroup analysis to synthesize DDL methods published in the last three to four years. Hence, the study is qualitative in nature. By undertaking the review and analysis approach mentioned earlier, our main objective is to enable researchers to get a general idea of the state of the DDL methods and their application in certain domains without increasing the needed time to synthesize and analyze the gathered information and report on findings. Furthermore, to offset the shortcoming of rapid review, we use an alternative synthesis method. While meta-analysis is highly desirable for comparing studies and driving new findings, acceptable synthesis methods, such as structured reporting with tabulation and visual displays and subgroup analysis, are better alternatives to the precluded meta-analysis when there is a concern about missing studies and statistical heterogeneity or simply heterogeneity of studies, i.e., variability among studies [22]. Additionally, our search strategy increased search specificity at the expense of search comprehensibility. We are limiting the search scope to include a relatively small set of publications while excluding others, which may result in excluding other relevant studies and selection bias. However, using a systematic review later to verify the critical outcomes of this survey can help address these issues [23].

To gather relevant and specific studies, we searched three repositories, including Howard University Libraries (accessed on 5 December 2021) (https://founders.howard.edu/using-the-libraries/), IEEEXplore (accessed on 5 December 2021) (https://ieeexplore.ieee.org/Xplore/home.jsp), and Google Scholar (accessed on 31 December 2021) (https://scholar.google.com/). We used the following search strings: data augmentation for imbalanced learning, data generation for adversarial learning, data augmentation using generative models, and data generation using generative models. In addition, we used the publication date and domain of application as our selection criteria to increase the specificity of the search. In particular, we filtered the results by publication date to consider only publications from 2021–2018. In addition, we sorted the publications by relevance using the IoT, CPS, IDS, network, and security domains of applications while excluding other domains. Finally, we conducted our analysis using structured reporting, which we augmented with tabulation and visual display methods and subgroup analysis to compare methods based on their evaluation results using the type of the approach, for example, data augmentation, data generation, or both. In summary, we conducted the following for data analysis and information synthesis:

1. First, we broadly classified the surveyed DDL methods based on the class of learning into imbalanced, adversarial, and non-adversarial (normal) learning.
2. Second, we considered methods employing more than one class of learning and those using more than one level of traffic (i.e., flow-level and packet-level), which we classified as hybrid data-generation methods.
3. Third, we considered the type of approach (e.g., data generation, data augmentation, and data imputation), application domain, and publication year.
4. Fourth, we used the type of approach to create subgroups (e.g., data augmentation and data generation), for further analysis and comparison
5. Finally, we compared the methods for each subgroup based on data quality, prediction accuracy, and performance.

In addition to imbalanced learning, non-adversarial or normal learning, and adversarial learning, we further considered other forms of learning to classify the DDL methods into conditional adversarial learning, transfer learning, statistical learning, exploiting learning, and deceptive learning. Finally, we defined the learning forms mentioned above in Table 1 and detailed our findings using the classification scheme in Table 2 and subgroup analysis based on the type of approach to reporting on the results of the evaluations in Table 3. Finally, we summarized the advantages and disadvantages of these methods in Table 4.

Table 1. Learning Classes Descriptions.

Learning Class	Description
Adversarial Learning	Train the model how to distinguish implausible data from plausible data to protect the system from inadvertently deceptive or misleading behavior
Conditional Adversarial Learning	Similar to adversarial learning, except that the learning happens in a conditional setting to create a general model where the generator and discriminator are conditioned on any auxiliary information such as class label, i.e., the model learns the loss and conditions the output of the system on its input [24,25]
Deceptive Learning	Train the model to modify adversarial examples to make them undetectable or evasive
Exploiting Learning	Train the model to exploit the generated data by one method to generate data using a different method, for example, generate synthetic data using the Monte Carlo method and then augment the synthesized data using an adversarial learning method
Imbalanced Learning	The model learns in the presence of skewed data distribution
Non-adversarial Learning	Train the model to generate benign examples indistinguishable from original benign examples to create new data, extend existing data, or compensate missing data
Statistical Learning	Train the model using statistics and functional analysis to make a prediction (Generative models are a type of statistical models)
Transfer Learning	Train the model to transfer its knowledge from a domain with adequate training data to other different but similar domain with inadequate or no training data

Table 2. Classification of Data-driven Learning Methods.

Method	Class **	Type *	Domain	Paper	Year
GAN-RF	IL	DAU	Network-based Intrusion Detection	[8]	2021
GAN-2CNN	IL	DAU & DG	Network-based Intrusion Detection	[26]	2021
IDSGAN	AL	DG	Network-based Intrusion Detection	[27]	2021
Bidirectional GAN	AL & NAL	DG	IoT-based Intrusion Detection	[17]	2021
G-IDS	IL	DAU & DG	Cyber-physical Systems	[28]	2020
GAN-SMOTE	IL	DAU & DG	Host-based Intrusion Detection	[12]	2020
AC-GAN	IL	DG	Smart Home-based Intrusion Detection	[9]	2020
GAN-AE	NAL	DG, DAG, & PP	IoT-based, Anomaly-based Detection	[29]	2020
IoT Sequential GAN	NAL	DAU	Predictive Maintenance/ IoT-based Household Energy	[10]	2020
GAN-based DA	AL & TL	DG & DAD	Adversarial Domain Adaptation	[30]	2020
PAC-GAN	NAL	DG & DIM	Traditional Communication Networks	[14]	2019
WGAN-GP	NL	DG	Traditional Communication Networks	[11]	2019
SynGAN	AL	DG	Network-based Intrusion Detection	[31]	2019
AdvGAN	CAL	DG	Defensive Security Adversarial Training	[32]	2019
NID Framework	IL, AL, SL, & EL	DAU & DG	Network-based Intrusion Detection	[33]	2019
Deceptive GAN	DL	DG	Offensive Security Adversarial Training	[34]	2018

* DAD: data adaptation, DAG: data aggregation, DAU: data augmentation, DG: data generation, DIM: data imputation, PP: privacy preservation; ** AL: adversarial learning, CAL: conditional adversarial learning, and DL: deceptive learning, IL: imbalanced learning; ** NAL: non-adversarial learning, NL: normal learning, SL: statistical learning, TL: transfer learning.

Table 3. Subgroup Analysis Based on the Evaluation Results of Data-driven Learning Methods.

Method	Type *	Data Quality	Predictability	Performance	Paper	Year
GAN-RF	AUG	No std measures	f-score > 95%	Superior/Good	[8]	2021
IoT Sequential GAN	AUG	Output layer score	f-score ≈ 60%	Variable	[10]	2020
IDSGAN	DG	No std measures	DR ≈ 0 and EIR > 99%	Good/Robust	[27]	2021
Bidirectional GAN	DG	Duration distribution	TPR 82%/FPR 0.02%	Promising	[17]	2021
AC-GAN	DG	No indication	f-score = 97%	Good	[9]	2020
WGAN-GP	DG	Euclidean distances (ED) and quality tests (QT)	ED between 0.02–0.14 and QT = 100%	Good except N-WGAN-GP	[11]	2019
SynGAN	DG	Quality benchmark and RMSE	RMSE = 0.10 and AUC = 75%	Good	[31]	2019
AdvGAN	DG	Attack success rate (ASR)	Accuracy 92.76% and ASR > 90%	Promising (Runtime < 0.01 s)	[32]	2019
Deceptive GAN	DG	IPS	Unblocking actions 63.42%	Promising	[34]	2018
GAN-2CNN	AUG & DG	f1-score ≈ 98% and 75%	f1-score > 92%	Outperforming	[26]	2021
G-IDS	AUG & DG	DR threshold	f1-score at least up to 91%	Good	[28]	2020
GAN-SMOTE	AUG & DG	No std measures	AUC > 0.97	Slightly reliable	[12]	2020
NID Framework	AUG & DG	No std measures	f1-score > 92%	Outperforming	[33]	2019
GAN-based DA	DG & DAD	No std measures	f-score > 88% and 83%	Outperforming	[30]	2020
GAN-AE	DG, DAG, & PP	No std measures	f1-score = 96%	Outperforming	[29]	2020
PAC-GAN	DG & DIM	Success rate (SR) and byte error	SR 99% and 88%	Promising	[14]	2019

* DAD: data adaptation, DAG: data aggregation, DAU: data augmentation, DG: data generation, DIM: data imputation, PP: privacy preservation.

Table 4. Advantages and Disadvantages of Data-driven Learning Methods

Method	Advantages	Disadvantages
GAN-RF	No overfitting and overlapping	No metrics to assess the data quality
IoT Sequential GAN	1D sequential data generation	Computation overhead and performance variation
IDSGAN	Adversarial training for IDS	Focus on blackbox attacks only
Bidirectional GAN	Compliant bidirectional flow generation	Small IoT traffic dataset, two features only to characterize network data, and no modeling of multimodal duration distributions or consideration of packet-level traffic
AC-GAN	Conversion of network data to images	Complex and computationally expensive due to data preprocessing prior to data generation
WGAN-GP	Learn internal dependencies between attributes without modeling additional knowledge, the ability to transform the heterogeneous flow	Suitable for generating single flow-based and new evaluation methods to assess the data quality network traffic only—In addition, the numeric transformation of IP addresses approach does not yield high quality data
SYNGAN	Emulation of real-world network attack mutations	Focused on generating only one type of attack (DDoS)

Table 4. *Cont.*

Method	Advantages	Disadvantages
AdvGAN	Generates perturbations for any input after training the feed-forward network without accessing the model, thus accelerating adversarial training	Complex framework using dynamic distillation to train the model for the blackbox attacks
Deceptive GAN	Self-adapting malware, self-adapting IPS, short training time without a need for a large amount of data	Works in flow-level without considering the packet-level, in addition to high communication overhead
GAN-2CNN	2D imagery representation of unseen 1D network attacks, simulating of unknown attack, and limited overfitting, overlapping, and noise	Complex methodology and implementation, different performance based on a variety of attacks in the dataset, and no detailed information on the structure and characteristics of attacks
G-IDS	Data from different sources collected and stored continuously and in parallel in a database and stabilization modeling during model training	Time complexity and computation overhead due to data processing and verification
GAN-SMOTE	Generation of system call traces for attacks on OS	Dataset limited to Linux OS call traces
NID Framework	Incorporation of adversarial learning with statistical and exploiting learning-based data augmentation and modeling feature distribution of network data	Computational complexity and high training time
GAN-based DA	Domain adaptation and knowledge transfer from sufficient data domain to small data domain and classification accuracy with minimal data	Requirements to use source and target datasets has privacy implications
GAN-AE	Data aggregation, privacy preservation, global learning of the data distribution, and minimum communication overhead	Scalability and inability of the local model to learn the diverse benign patterns from other networks
PAC-GAN	Realistic GAN-based flow at the IP packet level and IP packets to image-based matrix conversion	Communication overhead, transmission success accuracy, and low performance for mixed traffic generation

5. DDL Methods for Imbalanced Learning

The study in [8] suggested that most approaches that employ methods other than GAN suffered from data loss or overfitting and proposed the use of GAN to solve the data imbalance instead of resampling and SMOTE techniques to avoid overfitting caused by resampling and class overlapping or noise caused by SMOTE. The GAN generated virtual data similar to the minority class of the imbalanced data. The authors used the balanced data generated by the GAN, which solved the problem of overfitting and overlapping by specifying the desired resampling rate, to train an anomaly-based detection model based on the random forest (RF) method by increasing the weight of the minority attack class in the intrusion detection evaluation dataset (CICIDS). The GAN-based data augmentation method using resampling boosted the rare classes of the CICIDS 2017 dataset, which constituted less than 0.1% of the dataset, by generating 10,000 data of Bot, infiltration, and heartbleed. The batch size or the number of data learned at a time is 10 for Bot, 1 for the remaining two classes due to tiny data size (less than 30), and 20 for the epoch. The study compared the performance of the GAN-RF model, Single-RF model, and SMOTE-RF model using accuracy, precision, recall, and f-score. The GAN with Random Forest algorithm (GAN-RF) model used GAN for data resampling and RF for classification, standalone Random Forest algorithm (Single-RF) used RF for classification only, and SMOTE with Random Forest algorithm (SMOTE-RF) used SMOTE for data resampling and RF for classification. The GAN-RF performed better than the Single-RF and SMOTE-RF. Using the average score, GAN-RF had an accuracy of 99.83% and f-score of 95.04% compared to 99.19% accuracy and 87.79% f-score for the Single-RF model and 99.51% accuracy and 88.16% f-score for SMOTE-RF.

In addition to augmenting data by producing more examples to balance the minority class examples in the dataset, the GAN can simulate new unforeseen attacks. For example, the authors in [26] used GAN to augment network traffic represented using imagery to train a Convolutional Neural Network (CNN)-based intrusion detection model, and to simulate unforeseen attacks, we refer to this method as GAN 2D imagery CNN or GAN-2CNN for simplicity. However, the two-dimensional image of network flow, produced using two-dimensional mapping techniques, suffered from the unequal representation of normal and abnormal examples. The GAN addressed the imbalanced imagery issue by generating new images of unforeseen attacks, and the CNN classified the 2-D imagery, leading to better predictive accuracy for the GAN-2CNN model. The GAN-based imagery data augmentation method trained the auxiliary classifier GAN (AC-GAN), where it used the generator of the AC-GAN to create new synthetic attacks' images and balance the training dataset. A variant of GAN, AC-GAN, takes a class label and noise as input and generates images [35]. The employed AC-GAN's generator created fake images from a 100-dimensional input random noise vector of a uniform distribution and a two-dimensional one-hot label. The study analyzed the performance of the GAN-based imagery data augmentation using CICIDS17 and AAGM17 datasets with imbalanced traffic data and the full implementation of the model (referred to by the authors as MAGNETO, a supervised deep learning methodology for learning a robust intrusions model that deals with data imbalance). In addition, it measured the effectiveness of the data augmentation method compared to the SMOTE and adaptive synthetic (ADASYN), proposed by [36] data augmentation methods, and the effectiveness of training the 2D CNN using GAN-augmented data of varying balance sizes. Using a Variant of MAGNETO, i.e., MAGNETO with SMOTE (SMOTE-MAGNETO) and MAGNETO with ADASYN (ADASYN-MAGNETO), the MAGNETO with GAN (GAN-MAGNETO) outperformed the other variants on both datasets in terms of f-score and precision. However, GAN-MAGNETO exhibited a drop in recall, though negligible, using CICIDS17 compared to its performance using the AAGM17 dataset.

Imbalanced data can hinder the proper training of an AN-Intel-IDS and thus its performance. Publicly available datasets, such as the KDD-99 and CIDDS-001, are mostly imbalanced and often contain more ’normal’ examples than anomalous examples. The GAN-based IDS (G-IDS) in [28] for securing cyber-physical systems (CPS) addressed the issue of imbalanced data by generating more data to train the IDS, which is a multi-layer artificial neural network. It used the NSL KDD-99 to generate synthetic data that augmented the original data, thus, increasing the distribution of attack examples in the dataset. The proposed G-IDS framework consisted of four modules: database, IDS, controller, and synthesizer.

In addition to the generated synthetic data by the synthesizer module generator, the database module contained real-world intrusion detection data. The controller module decided whether to accept or reject pending data, i.e., synthetic data was data that had not been accepted or rejected by the controller. The GAN, which is part of the synthesizer module, generated the synthetic data. The synthesizer labeled the generated data as pending, due to the uncertainty of the GAN, before sending it to the database module. The authors used the controller module to evaluate the IDS module twice. First, they trained the IDS module on a hybrid dataset only, i.e., the combined original and synthetic data already accepted by the controller. Then, they trained the IDS module using a combination of the hybrid and pending datasets. The controller accepted or rejected the pending data based on the IDS performance. By measuring the detection rate for each data class (normal or attack) and comparing it to a pre-established performance threshold, the controller identified the weakly detected classes and sent the data examples to the synthesizer module to generate more examples. The process repeated until a satisfactory IDS performance was obtainable. Figure 1 shows the G-IDS framework, where P_H and P_p denoted IDS performance using hybrid and pending data, respectively. Comparing the performance of the G-IDS to a standalone IDS (S-IDS) or an IDS without a GAN using precision, recall, and F1 score as metrics, G-IDS performed better than S-IDS in terms of detection accuracy and stability on both the original and boosted datasets. The f-score of S-IDS was 70% to 78% using a 20% and 40% data increase, respectively, compared to G-IDS, which was 90% to 96% for a 20% and 40% data increase, respectively. However, G-IDS f-score dropped significantly to 60% for a 60% data decrease due to the G-IDS taking random noise as input. In terms of prediction accuracy, S-IDS performance suffered due to insufficient data. G-IDS generally had better performance and prediction accuracy; however, it is centralized and computationally expensive.

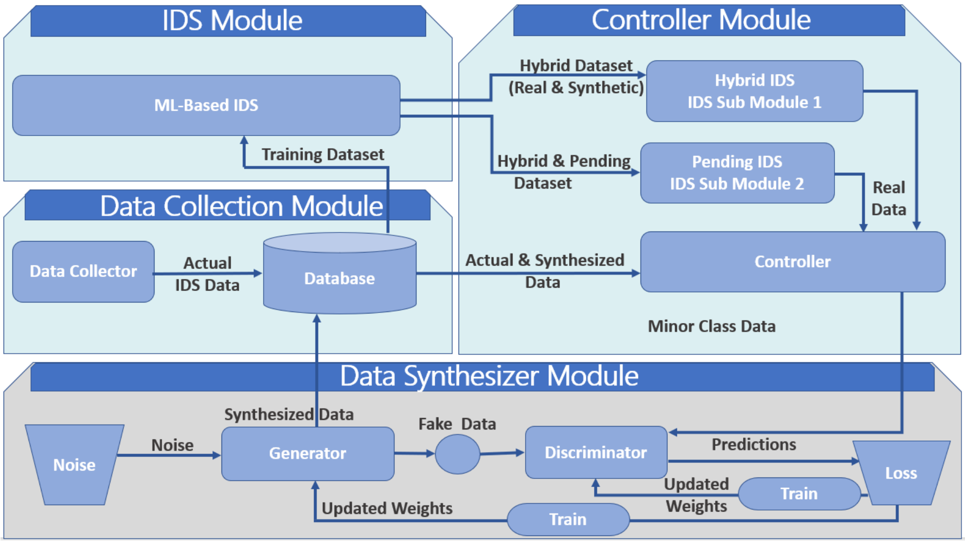

Figure 1. G-IDS Framework Block diagram (Recreated with permission from ref. [28]. Copyright 2022 Mohammad Ashiqur Rahman).

Unbalanced distribution of normal and attack examples in a dataset can lead to detection inaccuracies. Further, the detection accuracy of an IDS may vary based on the degree of class imbalance. The method in [12] addressed the issue of imbalanced learning using GAN-augmented data to train a supervised and unsupervised host-based intrusion detection system (HIDS), i.e., Support Vector Machines (SVM) and CNN, respectively. In addition to data augmentation using GAN, the author considered data oversampling using SMOTE to evaluate the performance of the GAN-based approach. The SMOTE-based approach over-sampled minority classes from unbalanced data, whereas the GAN-based approach generated the data (similar to the training dataset) itself. The dataset contained system-call trace data represented as a series of integer numbers mappable to system calls made on a Linux OS. Both approaches augmented the abnormal examples by creating new data that was invariably synthetic. The author applied both approaches to the pre-processed ADFA-LD dataset and then used SVM and CNN to classify process operation based on system call trace data into normal or malicious behavior. The GAN-based approach to data augmentation was slightly reliable compared to the SMOTE-based approach. In addition, models trained using augmented data had better classification accuracy than models trained using original data. In both cases, models that used GAN-based augmented data performed better. As the number of minority class examples increased by 30%, 50%, 70% and 100%, the classification accuracy and classification performance increased as well. In general, when using data augmentation, CNN performed better than SVM for largedata sizes, whereas SVM showed a better performance for moderate data sizes.

The number of attack examples in the smart home environment, is often smaller than the normal examples, thus creating data imbalance. Therefore, detecting intrusions in a smart home environment requires designing intelligent anomaly-based IDS capable of handling disproportions in the datasets. The authors in [9] proposed an embedded intrusion detection scheme on the smart homes edge nodes that exploited GAN to reduce the impact of disproportionate datasets, where normal examples are more frequent than attack examples, on the performance of the classifier. The authors used AC-GAN to generate synthetic data to balance the proportion of normal and attack examples in the UNSW-NB15 training dataset. The authors converted the network data into images prior to feeding the pre-processed data to the AC-GAN generator. In addition to a noise, the AC-GAN generator took the class label as input to generate synthesized data for the minority attack class. The authors then combined the synthesized data with the original data to train the classifier. The evaluation results showed that the proposed scheme, which included GAN-based data augmentation, improved the classifier precision for the minor attack class; the precision and recall of the anomaly detection was about 96% and 98%. However, when comparing the precision given the different categories of attacks, the precision of some of the attacks belonging to the majority class declined due to the low quality of the generated synthetic data.

6. DDL Methods for Adversarial Learning

6.1. GAN-Generated Regular Network Traffic

Training an anomaly-based intrusion detection system to detect intrusions in IoT environments is challenging due to the lack of sufficiently-large benign IoT data and the inability to collect IoT data from IoT devices directly due to high scalability and privacy restrictions. In addition, device disparity and activity scarcity make it harder to acquire reliable benign IoT data. The authors in [29] addressed these issues by proposing a data aggregation and privacy preservation hierarchical approach in which a GAN and an AE cooperated to reconstruct IoT benign data for training a global anomaly-detection IDS and set of local anomaly-detection IDS implemented at the local gateways. The hierarchical method used local GANs implemented at the local IoT networks to generate benign data and a global GAN to reproduce the aggregated benign data, which is double the size of the real data in the local IoT networks. Each local IoT network consisting of a set of local IoT devices and their generated data are aggregated at the global level using a centralized controller. The data generation occurred at the local GANs. First, the generator, which consisted of sequential layers, took a Gaussian Noise with random dimension size as input and generated a series of random outputs. Next, the discriminator combined the generated sample with the benign local data. Then, the generator and discriminator, which had symmetrical structures, were trained simultaneously. Finally, the data from the local generators were aggregated at the centralized AE to reproduce new benign data to train the global AE model that was double the size of the local networks' training data. Figure 2 shows the training map of the proposed hierarchical approach. The authors evaluated the proposed model GAN plus Autoencoder (GAN-AE) using the UNSW BoT-IoT dataset and two cases; the global model with all data and the local model with local data only. The GAN-AE model, global model, and local model outperformed four popular binary clustering approaches: one-class support vector machine (OSVM), isolation forest (IO), local outlier factor (LOF), and K-Means clustering. In particular, the GAN-AE, global model, and local model accuracy, precision, and recall were higher than 90%, with the global model outperforming the local model. However, the local model overfitting towards the local data was a drawback to the proposed approach, resulting in poor prediction accuracy of anomalies.

The study in [10] provided a tool to solve small data challenges in machine learning, where it is difficult and time-consuming to collect a representative amount of ground truth data. The authors used GAN to augment sequential IoT data, i.e., time-based sensor readings for predictive maintenance, and generate synthetic household energy consumption data. The generated data was subjectively similar to the original data. Before applying the data to the GAN, the authors first converted the one-dimensional sequential data into two-dimensional data by exploiting periodic behavior. This was necessary to exploit locality using the GAN and apply CNN methods. In doing so, they aimed at investigating if GAN with two-dimensional convolutions can generate one-dimensional sequential data to enable the use of sophisticated CNN methods such as sharing, pooling, and striding. However, the authors used a WGAN, adopted from the Keras WGAN implementation, instead of a deep convolutional GAN (DCGAN) due to vanishing gradients during the training of the DCGAN and the replacement of the discriminator's transfer function with a gradient penalty. The authors trained two of the WGANs; each WGAN was able to generate an energy consumption heatmap similar to the real data. To evaluate the quality of the GAN-generated data, they designed an evaluation workflow where they trained the generator with a subset of all data and used the generator output to train the classifier. The classifier training and evaluation involved using fake and real data, respectively. The data set contains two classes; a minority and majority class comprised of energy consumption data with and without swimming pool data. Further, the authors combined the WGAN with a convolutional neural network (CNN) and labeled data. The quantitative evaluation using labels revealed that it is possible to generate sequential data from small ground truth data or noise with fixed output size based on data with unique representation. In addition, the evaluation revealed an almost perfect classification for the majority class, where f-score was 0.95–1. However, the minority class f-score was 0.31 indicating poor classification.

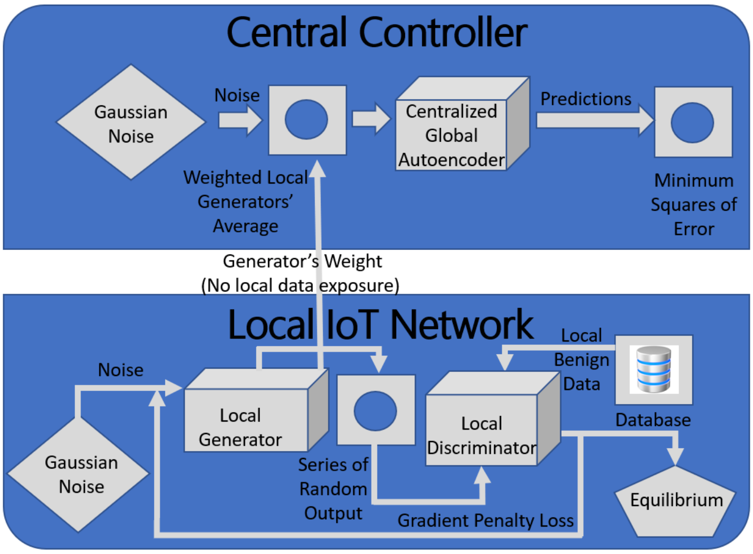

Figure 2. The Hierarchical Approach Training Map (Recreated with permission from ref. [29]. Copyright 2022 IEEE).

Historically, GANs are rooted in image recognition applications where they generate synthetic but realistic images from a given set of images as input. To generate realistic network traffic from GANs, the author in [14] proposed a convolutional neural network GAN traffic generator, named PAC-GAN, to generate packet-level network traffic that adheres to network standards and protocols. The proposed network traffic generator used an encoding scheme to convert and map network traffic data into images using image-based matrix representations. The PAC-GAN generated realistic variants of different types of network traffic, such as ICMP pings, DNS queries, and HTTP get requests to transmit through real networks by learning and manipulating the byte values of data packets. The encoding scheme encoded the GAN-generated network traffic and the training network traffic using the $n \times n$ matrix. Figure 3 shows each packet byte value mapped to an individual pixel in the matrix. The author measured the generator's performance using success rate and byte error as metrics, where the success rate is the number of a successfully sent packet to the total number of packets generated by the PAC-GAN and the byte error is the number of the incorrectly generated packet byte values averaged over all generated packets. A successfully sent packet is a dispatched packet over the internet that generated a valid response. The proposed generator achieved up to 99% success rate for individual traffic types and 88% for different traffic mixes.

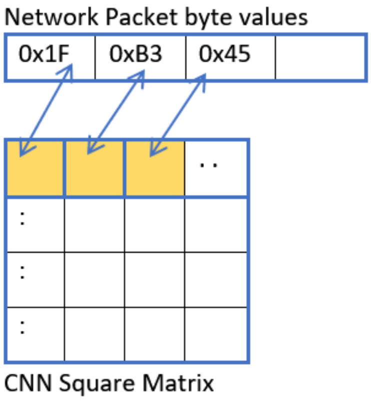

Figure 3. Mapping packet byte values into pixels (Recreated with permission from ref. [14]. Copyright 2022 IEEE).

Unlike the PAC-GAN method, the three synthetic flow-based network traffic generators based on the improved WGAN-GP proposed in [11] indirectly generated new flow-based network traffic based on the CIDDS-001 dataset by learning from the characteristics of previously collected network traffic to mimic the traffic flow. Further, they transformed the categorical attributes of the network traffic, such as protocol, IP addresses, and ports, into continuous attributes for processing by the GAN using three different pre-processing strategies: numeric transformation, binary transformation, and embedding transformation. First, the numeric transformation strategy transformed the IP address and the ports into numerical values. Second, the binary transformation strategy transformed the IP addresses, ports, bytes, and packets categorical values into binary attributes. Finally, the embedded transformation strategy transformed the categorical values of IP addresses, ports, bytes, and packets into vectors or embeddings in an m-dimensional continuous feature space. Three methods based on WGAN-GP, numeric WGAN-GP (N-WGAN-GP), binary WGAN-GP (B-WGAN-GP), and embedding WGAN-GP (E-WGAN-GP) implemented the numeric, binary, and embedding transformation, respectively. Given the processed flow, the WGAN-GP with the two time-scale update rule generated new flow-based network traffic whose quality was evaluated by the authors using a domain knowledge checks method. Further, the authors derived several properties to assess whether the generated data are realistic. The evaluation results indicated the ability of the E-WGAN-GP and B-WGAN-GP methods to generate realistic traffic. On the contrary, the N-WGAN-GP did not generate convincing, realistic data. A limitation of the WGAN-GP methods was that they generated single flows instead of sequences of flows.

6.2. GAN-Generated Network Intrusion Traffic

In general, there is a need to evaluate the robustness of intrusion detection systems such as the AN-Intel-IDS and improve their detection. One way to achieve this is by designing malicious traffic that can evade detection in real-world attack scenarios. To that effect, adversarial learning using GANs, such as the framework of GANs in [27], called IDSGAN, performed adversarial black-box attacks to deceive the IDS and to evade detection by generating new malicious traffic based on the original attack traffic. For example, the IDSGAN generated adversarial attacks based on the NSL-KDD by modifying the nonfunctional features in the original attack traffic that enabled it to deceive and bypass the IDS and launch an actual attack. The IDSGAN consisted of a generator, discriminator, and black-box IDS. Similar to [11], the authors used Wasserstein GAN to create the IDSGAN where the discriminator learned from a black-box IDS that mimicked a real IDS to ensure convergence and instability of the GAN. In addition, the generatorproduced a restricted form of adversarial malicious traffic by modifying limited features to ensure the validity of the generated adversarial traffic when launching a network attack in reality. shows the training of the IDSGAN framework. The authors evaluated the capacity and generality of IDSGAN against seven black-box IDS models they formed using different machine learning algorithms and trained using training sets based on the NSL-KDD dataset before the models generated adversarial attacks. Further, they used detection rate (DR) (number of correctly detected attacks divided by the number of all attacks) and evasion increase rate or EIR (one minus the adversarial detection rate divided by the original detection rate) as metrics. Further, they set the goal of the IDSGAN optimization such that a low detection rate and higher evasion increase rate were desirable. The evaluated IDSGAN had a good capacity generating adversarial malicious network traffic resulting in a very low detection rate for the black-box IDS. For non-modified adversarial malicious data, the IDSGAN maintained its evasion capacity.

Focusing on labeled data scarcity or sparsity and cost of data collection and labeling, the authors in [30] proposed the use of adversarial domain adaptation that leveraged GANs to transfer the knowledge gained from a domain with an adequate and existing training dataset to related but different domains with limited or no new training dataset, for example, transferring knowledge from the traditional network domain to the IoT domain. Figure 4 shows the architecture of the GAN-based domain adaptation (DA) or GAN-DA framework.

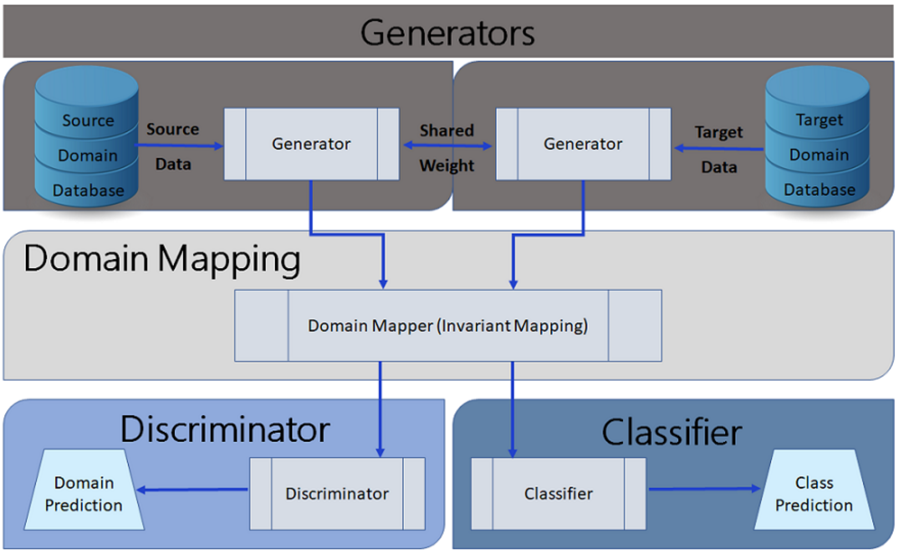

Figure 4. GAN-based Domain Adaptation (Recreated with permission from ref. [30]. Copyright 2022 Ankush Singla).

Apart from creating a domain invariant mapping between the two datasets, the proposed approach was feature-independent, i.e., it was applicable irrespective of the similarity or differences of the feature spaces of the source and target datasets. In addition, it was universal. Thus, it enabled the re-purposing of deep learning models in the target environment to operate in another environment that used similar data but different data representation using small labeled data from the target environment. Further, it reduced the large amount of labeled data required to train deep learning classifiers. The authors evaluated the proposed approach using publicly available network intrusion detection (NID) datasets and two scenarios where the source and target datasets had the same feature space (homogeneous DA where data was collected from devices using the same communication protocol). The source and target datasets had a different feature space (heterogeneous DA where data was collected from different types of devices using different protocols), respectively. Further, they used the same dataset, which they split into two parts to account for the source and target datasets, for the homogeneous scenario and two different datasets; one for the target and the other for the source, for the heterogeneous scenario. The proposed approach outperformed the base case, where the authors used the target dataset to train the deep learning model. The fine-tuning approach for a small dataset was better in terms of deep learning classification accuracy. The authors used the accuracy and f-score metrics when the source and target dataset had similar features and the f-score only when the features were different. As the number of samples increased, the GAN-DA approach performed better than the base and fined-tuned approaches. However, one issue with the proposed approach was the requirement to use the source and the target datasets, which is challenging to maintain when the source and target data collectors are different.

Similar to the IDSGAN framework, the synthetic GAN (SynGAN) framework in [31] used WGAN-GP to address the complexity and high quality of the generated synthetic network flow. However, unlike IDSGAN, which focused on generating synthetic normal flow, SynGAN applied the WGAN-GP to generate synthetic network attacks using NSL-KDD and CICIDS2017 public datasets. The authors used the two public datasets to measure the quality of the generated synthetic packets using a similarity index, i.e., the similarity between the synthesized and real network packets and the DDoS family of attacks to evaluate the SynGAN framework. The SynGAN framework consisted of three modules: the generator, the discriminator, and the evaluator. The authors used the Gradient Boosting as the evaluator. While the GAN discriminator differentiated between actual and artificial attacks, the evaluator differentiated between actual and artificial packets using a quality measure based on the root mean square error. The preliminary evaluation showed that the SynGAN framework could generate high-quality adversarial attacks with a root mean square error of 0.10, indicating that the proposed framework was incapable of distinguishing between actual and synthesized attacks.

The authors in [32] focused on efficiently generating adversarial examples with high perceptual quality using a GAN that accelerated adversarial training as defenses. They proposed adversarial GAN (AdvGAN), a conditional adversarial network similar in paradigm to GAN, that once trained instantly generated perturbations for any instances without the need to access the model. The generator of the AdvGAN was a feed-forward network that generated perturbations to create adversarial examples, whereas the discriminator ensured that the generated examples were realistic. Figure 5 shows the AdvGAN overall architecture. The authors evaluated the AdvGAN using both semi-whitebox and blackbox attack settings. There was no need to access the original target model after training the generator in the semi-whitebox attack settings. However, the authors trained a distilled model in blackbox attack settings and optimized the generator. Given different target models and state-of-the-art defenses, the AdvGAN had a higher success rate than other attacks using the MNIST and CIFAR-10 datasets for both attack settings. For example, using the MNIST dataset, the accuracy of the AdvGAN was 92.76% and 88.93% for the blackbox and semi-whitebox settings, respectively. Further, the generated attack instances were closer to the actual attack instances, and the generation process was efficient.

Rather than emulating malicious traffic to evade detection, a GAN can emulate normal traffic to bypass detection. In addition to generating adversarial malicious traffic to evade detection, a GAN can generate network traffic to mimic traffic of a legitimate application to evade detection, thus enabling the malware to adapt to the behavior of the IDS. To that extent, the authors in [34] used GANs, where the generators and discriminators were recurrent neural networks (RNN) to modify the network behavior of a real malware to mimic the behavior of Facebook chat network traffic. Their primary purpose was to create malware that can avoid detection by ML-based intrusion prevention systems (IPS) that exploit behavioral characteristics to detect malware. The authors used a threat model to demonstrate their approach that consisted of three components: detector, malware, and server. They deployed the GAN and malware in their laboratory local network, IPS in the router, and the server in the cloud. For each flow, the GAN modified the timing, duration, and request size. The adapted malware was tested if it was being blocked, and the GAN loss was fed back to the GAN. The malware and the blocking of the malware were real.

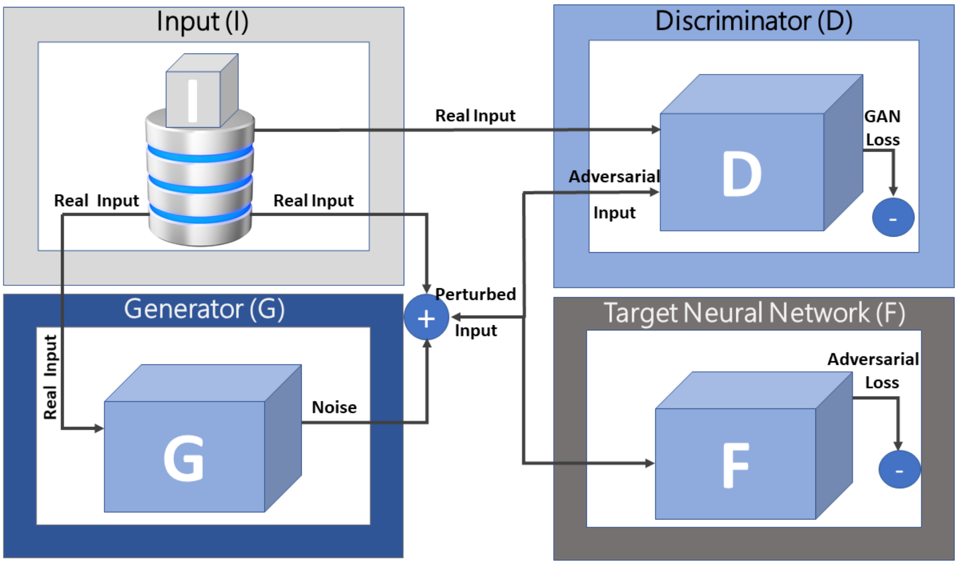

Figure 5. AdvGAN Architecture (Recreated with permission from [32]. Original is copyrighted with the International Joint Conferences on Artificial Intelligence (IJCAI), 2018. All rights reserved).

Using 217 network flows from normal traffic and training the GAN 400 times, the authors reported a drop in the blocking percentage to zero using enough numbers of epochs and a relatively small dataset, signaling a successful malicious action and the ability of the GAN to modify the malware traffic to avoid detection. The GAN was able to unblock 63.42% of the actions and allow 36.58% of the traffic to go undetected. However, the proposed method operated at the flow level rather than the packet level, and the improvement in the GAN performance was mainly attributed to additional training rather than data augmentation.

7. Hybrid DDL Methods

IoT traffic flow is bidirectional; therefore, methods for generating IoT synthetic data for training IoT intelligent IDS must consider bidirectional flow generation and the relationship between packet-level and flow-level features. The flow is composed of individual packets; thus, the packets' sizes are closely related to the flow duration. To this purpose, the author in [17] leveraged GAN to generate bidirectional flow that mimicked the bidirectional flow generated by actual IoT devices to train and test intelligent IoT IDS that used a set of sparse autoencoders; unsupervised neural networks. Unlike most of the surveyed synthetic data generation methods, which generated either packet-level features or flow-level features, the proposed generator created packet-level features while implicitly learning to comply with the flow-level characteristics to generate synthetic data that looks realistic. The flow-level features included packets' ordering, the total number of packets, and the total duration of the flow (total number of bytes). In contrast, features related to the packet-level included the packets' sizes. In general, packet-level features are describable using different fields of the network layer and the transport layer headers. The generated synthetic bidirectional flow consisted of a sequence of packets and their duration value. The trained generators using Autoencoder/WGAN with weight clipping(WGAN-C) model generated the sequence of packets. The trained mixture density networks (MDN), which took the generated packets sequence as input, determined their duration. The author used the WGAN to overcome the issues of GAN generating a sequence of categorical data, i.e., a sequence of packet sizes. The WGAN first converted the sequence of categorical data into a latent vector in a continuous space using the autoencoder and then trained the WGAN on the generated latent space to decode latent vectors into realistic sequences. Further, the author assessed the quality of the synthetic bidirectional flow by comparing the distribution of the duration of the synthetic bidirectional flow with that of the actual bidirectional flow and the sequence of packets sizes by using a Google Home Mini Show. The generated data are of quality if their duration is close to the duration of the real bidirectional flow. In both cases, the generated flow had a duration close to the real flow indicating the generated synthetic bidirectional flow was of high quality.

While the G-IDS framework in [28] focused on solving the imbalanced or missing data using adversarial learning, the network intrusion detection (NID) framework in [33] focused on solving the small and imbalanced dataset challenges using statistical learning and adversarial learning. The NID framework tackled both data scarcity and data imbalance by incorporating adversarial learning with statistical learning and exploiting learning using a data augmentation module (DA) consisting of a probabilistic generative model (PGM) and GAN. While the probabilistic model estimated the data feature distribution and generated synthesized intrusions using Monte Carlo methods, the deep generative neural network (DGNN) created high-quality intrusions by augmenting the synthesized data with actual data to provide high-quality training data. In addition, the PGM model initialized the DGNN, thus enabling it to converge on limited intrusion data. Figure 6 shows the structure of the DA module. The DA module enabled the NID framework to detect intrusions in small datasets. The authors used a GAN, which augmented the limited intrusion data, to adversarially train the DGNN and then evaluated the DA-enhanced NID using the KDD Cup 99 dataset against existing learning-based IDS, which included support vector machine (SVM), classical logistic regression (LR), and advanced DNN. The proposed NID framework outperformed the existing IDS, given accuracy, precision, recall, and f-score as metrics.

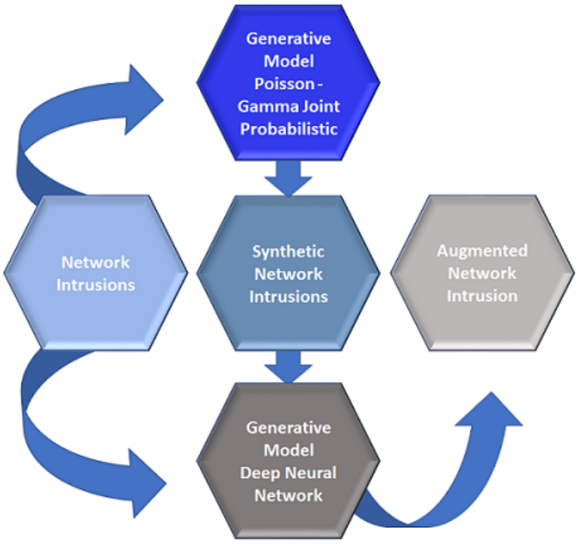
Generative Model Poisson - Gamma Joint Probabilistic
Network Intrusions
Synthetic Network Intrusions
Augmented Network Intrusion
Generative Model Deep Neural Network

Figure 6. DA Module Structure (Recreated with permission from ref. [33]. Copyright 2022 He Zhang).

8. Analysis and Discussion

In this section, we summarize our analysis and provide a classification of the surveyed methods and techniques. Our initial focus was to categorize the surveyed DDL methods and techniques into data augmentation and data generation based on the class of problem they are attempting to solve and into adversarial and non-adversarial learning based on their learning approach (see Table 2 for data-driven learning classification scheme). However, some techniques or methods tackle more than one problem, e.g., data augmentation and data generation, and employ two or more learning approaches, e.g., adversarial learning and non-adversarial learning. Table 2 lists several examples of these methods, such as the GAN-2CNN, G-IDS, GAN-SMOTE, GAN-based DA, PAV-GAN, and NID Framework, which address two classes of data problems, and the GAN-AE method, which considers solving three classes of data problems. The NID framework address two data problems and employ four different learning approaches: imbalanced learning, adversarial learning, statistical learning, and exploiting learning. Table 1 lists and describes these learning approaches and others mentioned in this study. In addition to data generation and data augmentation problems, some methods consider preserving data privacy, such as GAN-AE, and data adaptation, such as the GAN-based DA. Most of the methods have applications in network-based intrusion detection and fewer in IoT-based intrusion detection. Other methods have applications in cyber-physical systems security, defensive security, offensive security, and predictive maintenance of smart systems. While most of the methods focus on generating uni-directional flow-level or packet-level network traffic, the bidirectional GAN can generate bidirectional flow. Table 3 provides a subgroup analysis based on the evaluation results of the DDL methods and Table 4 summarizes the advantages and disadvantages of these methods.

Most of the studies presented in this paper focused on evaluating the data augmentation and generation methods using models trained on data augmented or generated by these methods compared to other state-of-the-art data augmentation and generation methods, ML and DL algorithms, or both. Further, most studies noted that the generated data, which resembled real data, is of good quality. However, few studies like the study in [11] assessed the quality and realism of the generated data using derived properties. Inevitably, there is a need to create standard metrics or develop a standard methodology to evaluate the quality of augmented and generated data. Furthermore, most studies evaluated the DDL methods using standard machine learning metrics, such as accuracy, precision, recall, and f-score.

On the other hand, few studies reported on the ROC metric, and others used specific metrics such as capacity, generality, detection rate, evasion increase rate, and percentage of unblocking actions. While these standard and specific metrics are essential to assess the performance of the proposed methods or approaches, there is a need to assess the sensitivity (true-positive rate) and specificity (false-positive rate) to indicate whether there was a significant improvement in the detection rate. In general, accuracy, precision, and recall metrics are not always a good indication of the models' performance.

Further, except for the study in [34], most of the studies assumed that the improvement in the performance of the proposed method was due to data augmentation or generation as opposed to the amount of training and model parameters. Data generation and augmentation methods invariably increase the computational cost. Nonetheless, some studies did not consider evaluating the proposed methods based on their execution time and computational overhead. Finally, most studies used static training, where models trained using publicly available data. Since public data are often static and prone to obsolesce, it is essential to consider dynamic learning and train models that consider the dynamic aspect of the data to augment or generate new data for imbalanced and adversarial learning.

9. Challenges and Open Research Issues

Generative models such as GANs are predominant in the field of image recognition. As such, they are more suited for discrete data generation. However, to extend their use beyond image recognition to other fields such as networking, GANs must have the capability to deal with categorical data such as IP addresses in addition to continuous data. Therefore, transforming network flow that contains categorical data into continuous value must occur before adversarial learning or imbalanced learning using generative models occurs. The authors in [11] proposed the use of three different prepossessing approaches for generating network flow. Others, such as the authors in [14], proposed a network encoding scheme to map network traffic from categorical format to image-based matrix representation.

Despite these transformation and mapping efforts, generating significant and realistic network traffic using cost-effective means remains challenging. When generating network traffic, it is essential to consider the traffic level. While some approaches generate network traffic at the flow level, others generate network traffic at the packet level. A better approach is to generate flow-based and packet-based traffic. To that extent, very few approaches, such as the Bidirectional GAN in [17], generated packet-level traffic and ensured that the generated traffic complies with the flow-level characteristics. Therefore, developing generative models that can exploit the relationship between flow-level and packet-level is an open research issue. Likewise, generating a sequence of flows instead of a single flow and generating bidirectional traffic for training IoT-based IDS is equally essential.

Evaluating GANs and assessing their data realism is challenging. However, the majority of GANs evaluation methods, both quantitative and qualitative, apply to image data [3]. Hence, developing methods for evaluating the performance of GANs and their variants for network data or non-imagery data are open for research. Another open issue to consider is defining metrics to evaluate GANs and their variants and assessing the authenticity and realism of the generated data. Current metrics exist at the individual GAN level, which makes it difficult to compare and assess different GANs [17]. Hence, there is a need to define standard metrics to assess the realism of the generated data. Furthermore, when considering generative models in networking, assessing data realism at a granular level, i.e., packet-level and flow-level, and assessing the generated data quality using comprehensive metrics are open questions. The generalization of generative models is an open research question. The initial intuition is to develop GANs that can adapt and transfer their knowledge from one domain of application to another similar yet different domain, e.g., from network-based intrusion detection to IoT-based intrusion detection. Moreover, developing generative models that consider bidirectional traffic generation is equally important.

10. Conclusions and Future Work

This paper presented an overview of various data augmentation and data generation methods for imbalanced and adversarial learning. The reason for augmenting data include but are not limited to small ground truth data, lack of attack data, preventive maintenance data, sensitive data, and data privacy. On the other hand, data generation is essential for adversarial learning, transfer learning, and deceptive learning. Further, the paper focused on the most recent research work published in the last three to four years and analyzed the findings using qualitative analysis. It used rapid review, structured reporting, and subgroup analysis, which narrowed the pool of selected publications to those which covered non-traditional ML/DL data augmentation and generation methods for training anomaly-based intelligent intrusion detection systems for detecting intrusions in traditional network and emerging fields of IoT, cybersecurity, and smart homes. Hence, it is limited in scope. This paper provided classification and a comparison of the reviewed methods and their implementing models and discussed their advantages and disadvantages to report findings. In addition, it introduced open issues and research challenges with a specific focus on categorical data mapping and transformation, evaluating and assessing generative models, generating packet-level and flow-level traffic, and bidirectional traffic. Most studies used standard machine learning metrics or domain-specific metrics to assess the augmented or generated data quality; there is a lack of standard data quality metrics and methodologies to assess the quality of the data. In addition, some studies were missing analysis on time and computational complexity, and communication overhead. In the future, we are planning to verify the outcome of this study using a systematic review and meta-analysis. Additionally, we plan to increase the scope of the systematic review to include network traffic transformation and mapping methods, and GANs variants for generating adversarial non-imagery examples.

Author Contributions: Conceptualization, G.A., J.W. and D.B.R.; methodology, G.A., J.W. and D.B.R.; investigation, G.A., J.W., D.B.R. and A.R.; resources, D.B.R. and A.R.; writing—original draft preparation, G.A. and J.W.; writing—review and editing, G.A., J.W., D.B.R. and A.R.; supervision, D.B.R.; project administration, D.B.R. and A.R.; funding acquisition, D.B.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the DoD Center of Excellence in AI and Machine Learning (CoE-AIML) at Howard University under Contract Number W911NF-20-2-0277 with the

U.S. Army Research Laboratory, by the Microsoft Research and by the US National Science Foundation under the grant number 1828811 and CNS/SaTC 2039583. However, any opinion, finding, and conclusions or recommendations expressed in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the funding agencies.

Acknowledgments: We thank the Multidisciplinary Digital Publishing Institute (MDPI) for the opportunity to publish our work and for their generous support.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Johnson, J.; Khoshgoftaar, T. Survey on deep learning with class imbalance. *J. Big Data* **2019**, *6*, 27. [CrossRef]
2. Mohammadi, B.; Sabokrou, M. End-to-End Adversarial Learning for Intrusion Detection in Computer Networks. In Proceedings of the 2019 IEEE 44th Conference on Local Computer Networks (LCN), Osnabrueck, Germany, 14–17 October 2019; IEEE: New York, NY, USA, 2019; pp. 270–273, ISSN 0742-1303.
3. Navidan, H.; Moshiri, P.F.; Nabati, M.; Shahbazian, R.; Ghorashi, S.A.; Shah-Mansouri, V.; Windridge, D. Generative adversarial networks (GANs) in networking: A comprehensive survey & evaluation. *Comput. Netw.* **2021**, *194*, 108–149.
4. Berg, A.; Felsberg, M.; Ahlberg, J. Unsupervised adversarial learning of anomaly detection in the wild. In Proceedings of the 24th European Conference on Artificial Intelligence-ECAI 2020, Santiago de Compostela, Spain, 31 August–2 September 2020; IOS Press: Amsterdam, The Netherlands, 2020; pp. 1002–1008.
5. Wong, S.; Gatt, A.; Stamatescu, V.; McDonnell, M. Understanding Data Augmentation for Classification: When to Wrap? In Proceedings of the 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA), Gold Coast, QLD, Australia, 30 November–2 December 2016; IEEE: New York, NY, USA, 2016; pp. 1–6, ISSN 978-1-5090-2896-2.
6. Ekbatani, K.; Pujol, O.; Segui, S. Synthetic Data Generation for Deep Learning in Counting Pedestrians. In Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods-ICPRAM, Porto, Portugal, 24–26 February 2017; Science and Technology Publishing: Setubal, Portugal, 2017; pp. 318–323, ISSN 2184-4313.
7. Seffens, W.; Evans, C. Machine Learning Data Imputation and Classification in a Multicohort Hypertension Clinical Study. *Bioinform. Biol. Insights* **2015**, *9*, 43–54. [CrossRef] [PubMed]
8. Lee, J.; Park, K. GAN-based imbalanced data intrusion detection system. *Pers. Ubiquit. Comput.* **2021**, *25*, 121–128. [CrossRef]
9. Yuan, D.; Ota, K.; Dong, M.; Zhu, X.; Wu, T.; Zhang, L.; Ma, J. Intrusion detection for smart home security based on data augmentation with edge computing. In Proceedings of the ICC 2020, 2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020; IEEE: New York, NY, USA, 2020; pp. 1–6, ISSN 1938-1883.
10. Tschuchnig, M.E.; Ferner, C.; Wegenkittl, S. Sequential IoT data augmentation using generative adversarial networks. In Proceedings of the ICASSP 2020, 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; IEEE: New York, NY, USA, 2020; pp. 4212–4216, ISSN 2379-190X.
11. Ring, M.; Schlör, D.; Landes, D.; Hotho, A. Flow-based network traffic generation using generative adversarial networks. *Comput. Secur.* **2019**, *82*, 156–172. [CrossRef]
12. Kim, K. GAN based augmentation for improving anomaly detection accuracy in host-based intrusion detection systems. *Int. J. Eng. Res. Technol.* **2020**, *13*, 3987. [CrossRef]
13. Leevy, J.L.; Khoshgoftaar, T.M.; Bauder, R.A.; Seliya, N. A survey on addressing high-class imbalance in big data. *J. Big Data* **2018**, *5*, 42. [CrossRef]
14. Cheng, A. PAC-GAN: Packet generation of network traffic using generative adversarial networks. In Proceedings of the 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 17–19 October 2019; IEEE: New York, NY, USA, 2019; pp. 0728–0734, ISSN 2644-3163.
15. Yin, C.; Zhu, Y.; Liu, S.; Fei, J.; Zhang, H. An enhancing framework for botnet detection using generative adversarial networks. In Proceedings of the 2018 International Conference on Artificial Intelligence and Big Data (ICAIBD), Chengdu, China, 26–28 May 2018; IEEE: New York, NY, USA, 2018; pp. 228–234.
16. Purser, J.L. Using Generative Adversarial Networks for Intrusion Detection in Cyber-Physical Systems. Master's Thesis, Naval Postgraduate School, Monterey, CA, USA, 2020.
17. Shahid, M.R. Deep Learning for Internet of Things (IoT) Network Security. Ph.D. Thesis, Institut Polytechnique de Paris, Palaiseau, France, 2021.
18. Di Mattia, F.; Galeone, P.; De Simoni, M.; Ghelfi, E. A survey on GANs for anomaly detection. *arXiv* **2021**, arXiv:1906.11632.
19. Zhang, J.; Li, C. Adversarial examples: Opportunities and challenges. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 2578–2593. [CrossRef] [PubMed]
20. Chika, Y.-B.; Ogban-Asuquo, U. A review of generative adversarial networks and its application in cybersecurity. *Artif. Intell. Rev.* **2020**, *53*, 1721–1736.
21. Krawczyk, B. Learning from imbalanced data: Open challenges and future directions. *Prog. Artif. Intell.* **2016**, *5*, 221–232. [CrossRef]

22. Higgins, J.; Thomas, J.; Chler, J.; Cumpston, M.; Li, T.; Page, M.; Welch, V. *Cochrane Handbook for Systematic Reviews of Interventions*; Version 6.2 (Updated February 2021); John Wiley & Sons: Chichester, UK, 2021. Available online: www.training.cochrane.org/handbook (accessed on 26 December 2021).
23. Mikolajewicz, N.; Komarova, S. Meta-Analytic Methodology for Basic Research: A Practical Guide. *Front. Physiol.* **2019**, *10*, 203. [CrossRef] [PubMed]
24. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. *arXiv* **2018**, arXiv:1611.07004.
25. Mirza, M.; Osindero, S. Conditional generative adversarial nets. *arXiv* **2014**, arXiv:1411.1784.
26. Andresini, G.; Appice, A.; De Rose, L.; Malerba, D. GAN augmentation to deal with imbalance in imaging-based intrusion detection. *Future Gener. Comput. Syst.* **2021**, *123*, 108–127. [CrossRef]
27. Lin, Z.; Shi, Y.; Xue, Z. IDSGAN: Generative adversarial networks for attack generation against intrusion detection. *arXiv* **2021**, arXiv:1809.02077.
28. Shahriar, M.H.; Haque, N.I.; Rahman, M.A.; Alonso, M., Jr. G-IDS: Generative adversarial networks assisted intrusion detection system. *arXiv* **2020**, arXiv:2006.00676.
29. Zixu, T.; Liyanage, K.S.K.; Gurusamy, M. Generative adversarial network and auto encoder based anomaly detection in distributed IoT networks. In Proceedings of the GLOBECOM 2020, 2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020; IEEE: New York, NY, USA, 2020; pp. 1–7, ISSN 2576-6813.
30. Singla, A.; Bertino, E.; Verma, D. Preparing network intrusion detection deep learning models with minimal data using adversarial domain adaptation. In Proceedings of the 15th ACM Asia Conference on Computer and Communications Security, Ser. ASIA CCS '20, Taipei Taiwan, 5–9 October 2020; ACM: New York, NY, USA, 2020; pp. 127–140, ISBN 978-1-4503-6750-9/20/10.
31. Charlier, J.; Singh, A.; Ormazabal, G.; State, R.; Schulzrinne, H. SynGAN: Towards generating synthetic network attacks using GANs. *arXiv* **2019**, arXiv:1908.09899.
32. Xiao, C.; Li, B.; Zhu, J.Y.; He, W.; Liu, M.; Song, D. Generating adversarial examples with adversarial networks. *arXiv* **2019**, arXiv:1801.02610.
33. Zhang, H.; Yu, X.; Ren, P.; Luo, C.; Min, G. Deep adversarial learning in intrusion detection: A data augmentation enhanced framework. *arXiv* **2019**, arXiv:1901.07949.
34. Rigaki, M.; Garcia, S. Bringing a GAN to a knife-fight: Adapting malware communication to avoid detection. In Proceedings of the 2018 IEEE Security and Privacy Workshops (SPW), IEEE Symposium on Security and Privacy Workshops (SPW), San Francisco, CA, USA, 24 May 2018; IEEE: New York, NY, USA, 2018; pp. 70–75.
35. Odena, A.; Olah, C.; Shlens, J. Conditional image synthesis with auxiliary classifier GANs. In Proceedings of the 34 th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; PMLR: Cambridge, MA, USA, 2017; pp. 2642–2651.
36. Habibo, H.; Yang, B.; Garcia, E.; Shutao, L. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–6 June 2008; IEEE: New York, NY, USA; 2008; pp. 1322–1328, ISSN 2161-4407.