Applying Generative Machine Learning to Intrusion Detection: A Systematic Mapping Study and Review

JAMES HALVORSEN, School of EECS, Washington State University, Pullman, United States
CLEMENTE IZURIETA, School of Computing, Montana State University, Bozeman, United States and Idaho National Laboratory, Idaho Falls, United States
HAIPENG CAI, School of EECS, Washington State University, Pullman, United States
ASSEFAW GEBREMEDHIN, School of EECS, Washington State University, Pullman, United States

Intrusion Detection Systems (IDSs) are an essential element of modern cyber defense, alerting users to when and where cyber-attacks occur. Machine learning can enable IDSs to further distinguish between benign and malicious behaviors, but it comes with several challenges, including lack of quality training data and high false-positive rates. Generative Machine Learning Models (GMLMs) can help overcome these challenges. This article offers an in-depth exploration of GMLMs' application to intrusion detection. It gives (1) a systematic mapping study of research at the intersection of GMLMs and IDSs, and (2) a detailed review providing insights and directions for future research.

CCS Concepts: • **Computing methodologies → Neural networks**; *Supervised learning by classification*; • **Security and privacy → Intrusion detection systems;**

Additional Key Words and Phrases: Generative Models, Penetration Testing, Unbalanced Datasets, Cyber Alert Generation, Flow Generation, Evaluation Metrics

ACM Reference Format:
James Halvorsen, Clemente Izurieta, Haipeng Cai, and Assefaw Gebremedhin. 2024. Applying Generative Machine Learning to Intrusion Detection: A Systematic Mapping Study and Review. *ACM Comput. Surv.* 56, 10, Article 257 (June 2024), 33 pages. https://doi.org/10.1145/3659575

1 INTRODUCTION

The modern cyber threat landscape is complex and costly. Although the general public's awareness of magnitude is rather limited, cyber attacks pose a significant threat to society's economic infrastructure [19]. In 2020, it was estimated that the global cost of cyber attacks had reached nearly $1 trillion [95], with ransomware representing the fastest growing cyber threat. Several recent attacks on critical energy infrastructure around the world have demonstrated that the threat posed by cyber attacks could become more than just a monetary one. Examples of such attacks include the Stuxnet worm [53], which attacked uranium enrichment facilities in Iran in 2010, and

Authors' Contact Information: James Halvorsen, School of EECS, Washington State University, Pullman, Washington, United States; e-mail: james.halvorsen@wsu.edu; Clemente Izurieta, School of Computing, Montana State University, Bozeman, Montana, United States and Idaho National Laboratory, Idaho Falls, Idaho, United States; e-mail: clemente.izurieta@montana.edu; Haipeng Cai, School of EECS, Washington State University, Pullman, Washington, United States; e-mail: haipeng.cai@wsu.edu; Assefaw Gebremedhin, School of EECS, Washington State University, Pullman, Washington, United States; e-mail: assefaw.gebremedhin@wsu.edu.

ACM 0360-0300/2024/06-ART257
https://doi.org/10.1145/3659575

a variant of the BlackEnergy malware [55], which was used to disrupt the services of Ukrainian power companies in 2015.

To combat cyber attacks, **Intrusion Detection Systems (IDSs)** are employed to monitor hosts and networks for signs of intrusion and malicious activity [56]. There are two dominant types of IDSs: *signature based* and *anomaly based*. The basic idea in signature-based IDSs is to work with a list of known indicators of compromise and searching for those indicators in new contexts. Such an approach can detect attackers using older, well-established techniques but cannot detect attackers exploiting undisclosed vulnerabilities. Anomaly-based IDSs work with the idea of having a model of known correct behavior and search for deviations from that behavior. Such approaches can detect both known and unknown types of attacks, but they may have more false positives generating alerts for normal activity.

Anomaly-based IDSs can greatly benefit from using machine learning to develop a more complex model to classify behavior as malicious or benign. This potential carries with it overcoming challenges associated with using machine learning approaches in general. Of particular note are performance-related issues (i.e., high false-positive rates) and issues related to data access (i.e., the lack of quality security datasets).

Generative Machine Learning Models (GMLMs) can offer a solution to both of these problems. Starting with the invention of **Variational Autoencoders (VAEs)** [49] in 2013 and that of **Generative Adversarial Networks (GANs)** [28] in 2014, there has been a significant increase in interest in both the methods for developing GMLMs and the application areas in which they can be employed. GMLMs have since enjoyed incredible success in various domains outside of intrusion detection, most notably in computer vision and artistic applications. Examples of such applications include generating images of faces [11] or producing poems or music [125].

Part of GMLMs' success in these domains can be attributed to the ease with which the products can be evaluated. In addition to objective evaluation with traditional, numeric-valued metrics, these artistic applications can be evaluated subjectively in terms of their aesthetic qualities. One such example is given by Cheng et al. [16], who evaluated several GAN models on the MNIST dataset (which consists of images of handwritten digits) using both quantitative and qualitative approaches. In the qualitative evaluation, some of the less realistic images were described as having "high distortion" or "incomplete," whereas the higher-quality images were described as "sharper."

Compared to generating art, generating cyber security data is a much more challenging problem domain. One of the great challenges at the moment is the lack of standardization of evaluation metrics. Unlike art, it is not intuitive to visually assess how realistic data intended for intrusion detection tasks may be. This leaves only objective metrics, which are myriad, and do not necessarily bestow the same confidence as being able to visually confirm that the results appear reasonable.

Driven by the potential GMLMs hold for major advancements in anomaly-based intrusion detection and the research needs for overcoming the aforementioned challenges around evaluation metrics, there is a growing body of work investigating the application of GMLMs to intrusion detection tasks. The works vary in their approach to the IDS tasks, but their primary contributions tend to focus on either solving issues related to performance or addressing issues related to data access.

The goal of this survey is to offer an in-depth exploration of the application of GMLMs to intrusion detection. Although survey papers on related topics exist, we are not aware of any prior work that focuses at the intersection of GMLMs and intrusion detection. Notable examples of existing surveys include the review by Yinka-Banjo and Ugot [124] of GANs in cyber security and the survey by Dutta et al. [24] of GANs in cyber security. Both works focus exclusively on GANs, whereas we consider GMLMs more broadly. Furthermore, we provide a detailed examination of

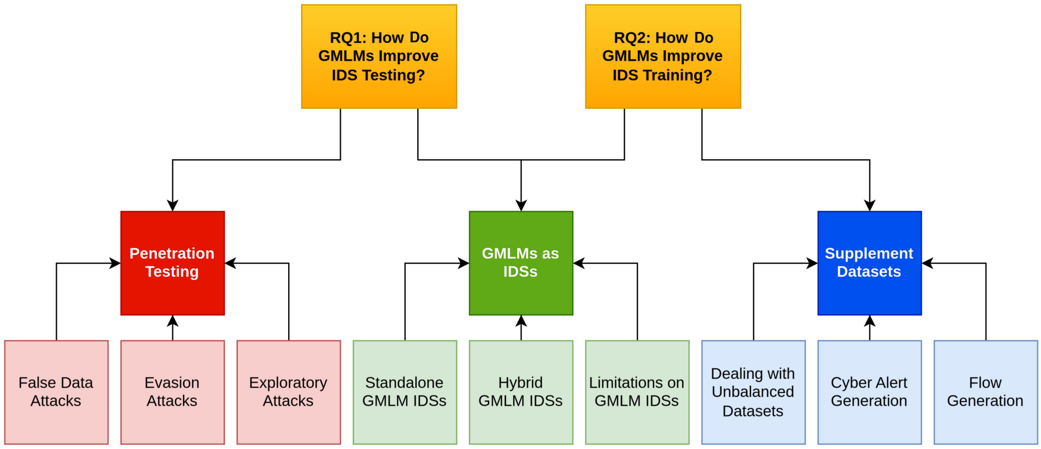


Fig. 1. Overview of applications of GMLMs to IDSs discovered in this mapping study. The top row shows RQ1 and RQ2, the research questions posed in the study; the middle row shows the application areas discovered by the study; and the bottom row shows the topics identified in the areas.

the subject of intrusion detection rather than cyber security in general. Finally, we employ a novel approach in our review and analysis.

Specifically, our article has a twofold purpose. The first purpose is to present a *systematic mapping study* [79] that establishes how GMLMs have been used in the domain of intrusion detection. A systematic mapping study typically provides a structure of the type of published research papers and reports in an area (e.g., software engineering or medical research) by categorizing the published works and often gives a visual summary, a map, of its results [79]. In our case, we conduct the mapping study to understand at a high level how GMLMs have been used in intrusion detection. The second purpose of our article is to provide a deeper analysis and synthesis of the works included in the mapping study. The end goal of this second purpose is to bring out insights, discuss strengths and weaknesses of methods employed, outline important challenges, and point out directions for future research.

Overview of Contributions and Structure of the Article

The article is structured reflecting the two purposes while also facilitating exposition.

Section 2 discusses survey papers on topics related to ours and explains how our work differs in nature and scope. To furnish the necessary background for the rest of the article, Section 3 gives a high-level review of different GMLMs and how they are used to generate data.

Section 4 describes the methodology used for including and excluding papers in our mapping study, provides relevant statistics on the papers selected, and presents the three main application areas of GMLMs in IDSs discovered by the mapping study. The three discovered areas are (1) *GMLMs for assisting with penetration testing*, (2) *GMLMs for supplementing IDS datasets*, and (3) *GMLMS as IDSs*. An overview of the three IDS application areas and nine specific topics identified under them (three in each area) is given in Figure 1. Expanding on the mapping study, Section 5 reviews *evaluation metrics* used in each of the three discovered application areas and gives our own results analyzing how the metrics have been used in the works surveyed in the study.

Addressing the second purpose of this article, the three application areas and associated nine specific topics are examined in detail in Section 6 (penetration testing), Section 7 (supplementing datasets), and Section 8 (GMLMs as IDSs). Sections 6 through 8 each contain three subsections corresponding to the three topics in the area and a subsection headed "Discussion" that synthesizes the works surveyed in the section. For example, the content of Section 6 is Background (Section 6.1), False Data Attacks (Section 6.2), Evasion Attacks (Section 6.3), Exploratory Attacks (Section 6.4), and Discussion (Section 6.5). Sections 7 and 8 are organized similarly.

We conclude the article in Section 9 with a summary and a list of open problems and directions for future work.

2 RELATED WORK

IDSs have a relatively long history, having first been proposed in 1980 [7]. Since then, several different approaches to intrusion detection have been developed. Lunt et al. [62] created the first host-based IDS that could learn a subject's behavior and detect deviations from it over time. Around the same time, Heberlein et al. [33] developed a similar work for behavior-based network intrusion detection. Modern approaches to intrusion detection typically use deep learning techniques to perform anomaly-based detection. Vinayakumar et al. [112] provide an example of this approach in their work with deep neural networks for intrusion detection on the KDDCup99 dataset.

Several prior works explore a portion of the relationship between GMLMs and intrusion detection. However, the topic is often as broad as machine learning models in general or explores only specific GMLMs, such as GANs. Yinka-Banjo and Ugot [124] provide a review of several different variants of GANs and discuss how they can be applied to the task of detecting adversarial attacks. Dutta et al. [24] consider GANs in security tasks more broadly and explore their use in other applications such as guessing passwords and breaking ciphers.

Where non-GMLMs are considered, there are many more works exploring intrusion detection tasks. Liu and Lang [59] provide an in-depth review of a wide variety of machine learning models that are used for intrusion detection tasks. GANs and autoencoders are briefly discussed among many other deep learning models. Haq et al. [32] survey a wide variety of machine learning approaches used for intrusion detection between the years of 2009 and 2014 and report on their relative performance in terms of detection rate. However, due to its age, it does not report on modern generative models.

Similar work is provided by Kishor Wagh et al.[50], who cover a number of models not discussed in the work of Haq et al. [32] and offer some analysis on future directions for intrusion detection, including the need for a standard evaluation dataset. A more recent work by Singh and Khare [94] discusses the more common datasets used for intrusion detection in greater depth, along with several issues including their lack of balance and recency. Further analysis on the challenges faced in intrusion detection, including those related to data, are discussed by Khraisat et al. [45].

Our work differs from these previous works by focusing on the use of GMLMs broadly (not just GANs) for intrusion detection and by presenting an in-depth analysis and synthesis. It discovers three application areas within intrusion detection where GMLMs are used and provides a systematic mapping study showing how they are applied to these tasks. Systematic mapping studies are a common approach used in medical research and in software engineering fields for developing classification schemes and a better understanding of a body of literature [79]. Some examples of previous mapping studies include the work by AcuÃśa et al. [2] on open source software development processes and the work of Penzenstadler et al. [78] on software engineering for sustainability. Our work differs from a traditional systematic mapping study in that it also provides a detailed analysis of the underlying challenges as well as the benefits and shortcomings of the methods employed. Due to the popularity of GANs in this domain, the majority of works surveyed are GANs. However, other GMLMs such as VAEs are considered as well.

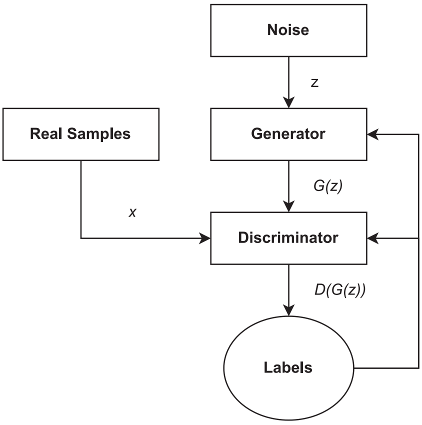


Fig. 2. Computation procedure and structure of GAN. Illustration reproduced from the work of Wang et al. [113].

3 OVERVIEW OF GENERATIVE MODELS

Generative models are machine learning models that are trained to generate random instances of some class of data. Until the invention of GANs [28] in 2014, generative models have had little success, due to the need to solve intractable probabilistic computations. In the past decade, however, several approaches to generative machine learning have been developed, which have been expanded beyond just GANs. This section reviews basic concepts behind some of these approaches, including the use of GANs, VAEs, and diffusion models.

3.1 Generative Adversarial Networks

In the GAN framework, two models are trained together in an adversarial process—a generative model and a discriminative model [28]. A common analogy used to describe the two is that the generative model is much like a team of counterfeiters, trying to produce fake currency, whereas the discriminative model is like the police, trying to identify the counterfeit currency. The generative model trains to become better at fooling the discriminative model, which in turn trains to become better at distinguishing between generated and real data.

The generator (G) and discriminator (D) are both typically implemented as some type of neural network, such as a deep convolutional neural network. During the training process, the discriminator is trained to maximize the probability of assigning the correct label both to real training samples and adversarial samples from the generator. Simultaneously, the generator is trained to minimize the likelihood that the discriminator will discriminate correctly. This process is illustrated in Figure 2.

The optimization process can be formulated as a minimax game on a value function $V(G, D)$ as shown in Equation (1). There, p_{data} represents the distribution of the actual training data, and p_z is a distribution of noise variables used as input to the generator.

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[log(D(x))] + \mathbb{E}_{z \sim p_z(z)}[log(1 - D(G(z)))] \quad (1)$$

Since the work of Goodfellow et al. [28] was published, a number of variants on the original GAN framework have emerged [113]. A commonly used variation is the **Wasserstein GAN (WGAN)**, developed by Arjovsky et al. [9] to improve the stability of learning compared to traditional GAN training. This training process was improved even further in the work by Gulrajani et al. [31] with the use of weight clipping. Other GAN variants include Bidirectional GANs (BiGANs), which train

ALGORITHM 1: Minibatch version of the original VAE algorithm presented in the work of Kingma and Welling [49].

```
$\theta, \phi \leftarrow$ Initialize Parameters
while $(\theta, \phi)$ are not converged do
    $X^M \leftarrow$ Random minibatch of $M$ datapoints (drawn from full dataset)
    $\epsilon \leftarrow$ Random samples from noise distribution $p(\epsilon)$
    $g \leftarrow \nabla_{\theta,\phi} \widetilde{\mathcal{L}}^M(\theta, \phi; X^M, \epsilon)$ (Gradients of minibatch estimator)
    $\theta, \phi \leftarrow$ Update parameters using gradients $g$ (e.g., using SGD or Adagrad)
end while
return $\theta, \phi$
```

an encoder that is an inverse function of the generator [23], and Energy-Based GANs (EBGANs), which view the discriminator as an "energy function" that can act as a cost function for the generator [128].

A number of variations of GANs have been tailored to specialized applications. Among these include the MidiNet of Yang et al. [118], which is a GAN that uses convolutional neural networks to form the base of its generator and discriminator to generate music. **Conditional GANs (CGANs)** may also be useful in generating specialized GANs, as in the case of the work of Zhang et al. [127] on image de-raining, which uses a CGAN to enforce a constraint that a de-rained image be indistinguishable from the ground truth image.

More recent applications of GANs include PolyGAN [77], a multi-CGAN designed for fashion synthesis, which can take images of models and clothing, and generate images of those models wearing the given clothing in different poses. CEGAN [98] is another important architecture, designed to improve existing GAN architectures for solving data imbalance issues. GANs are additionally capable of non-generative tasks, and a recent example of this is M3GAN [54], which is a GAN for time-series anomaly detection.

3.2 Variational Autoencoders

VAE is a framework for training two machine learning models, an encoder and decoder, that can be used for generative tasks. The final goal of producing an encoder and decoder model is similar to standard autoencoder frameworks, but the training process and mathematical basis for VAEs are much different [22].

In VAE, it is assumed that the input dataset was produced by some random process. A component of this process is the prior distribution $p_\theta(z)$, which has unknown parameters θ^* and latent variables $z^{(i)}$[49]. The VAE training process attempts to learn this parameter θ to learn the likelihood $p_\theta(x|z)$ to be used as a probabilistic decoder. It also learns a parameter ϕ for a reconstruction model $q_\phi(z|x)$, which is an approximation of the posterior distribution $p_\theta(z|x)$, which, in turn, is much more difficult to compute. This reconstruction model is used as the probabilistic encoder. Algorithm 1, which is reproduced from the original work [49], demonstrates how the parameters θ and ϕ are computed.

Once trained, the VAE's decoder can be used for generative tasks by providing it with a source of random noise, much like with a GAN's generator. The encoder, by contrast, converts data into a latent representation which can then be reconstructed with the decoder. A commonly used illustration for this architecture is displayed in Figure 3. Reconstructing data that has been encoded can result in some degree of loss, which can be used for anomaly detection as discussed in Section 8. Variants on the VAE algorithm can be used to extend their generative capabilities, such as conditional VAEs [96], which can be used to generate structured output.

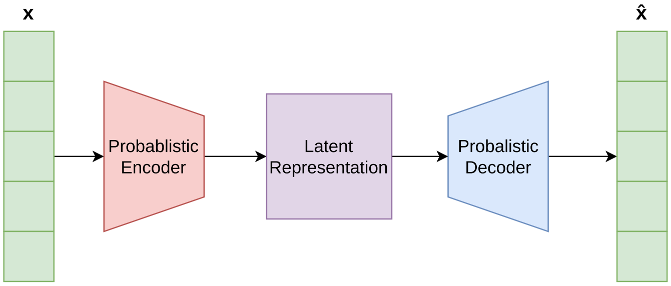


Fig. 3. VAE architecture. The original input data is given by the vector x, and the reconstructed data is given by the vector $\hat{x}$. This diagram is inspired by several other similar diagrams depicting the same process.

Recent innovations on the VAE architecture include **Channel-Recurrent VAEs (CR-VAEs)** [91], which incorporate LSTMs to process convolutional features and achieve higher-quality image modeling. Further iterations on CRVAEs are used in tasks neurodegenerative disease modeling [64]. Other recent innovations include NVAE [110], which allows VAEs to benefit from batch normalization.

3.3 Other Generative Models

Compared to GANs and VAEs, there are not as many works relating other GMLMs to IDS applications. The following are examples of approaches to generative machine learning that have been successful in other fields but need more research to demonstrate their applicability to intrusion detection.

Diffusion models are probabilistic generative models that are trained via a process that involves first corrupting the training data by adding increasing quantities of noise, then learning to reverse this process [34]. Thus far, they have had significant success in the field of computer vision [17] and image generation [13]. These models in particular have gained considerable attention due to their use in creating AI-generated artwork thanks to tools such as Stable Diffusion [84] and DALL-E 2 [80].

Deep autoregressive networks are a method of training deep autoencoders that are able to sample data through ancestral sampling [29]. Thus far, common application areas for autoregressive models have focused on the generation of sequential data. They have seen significant success thus far in large language models such as GPT-3 [15] and GPT-4 [74], as well as some image generator models such as PixelCNN [111].

Normalizing Flows are a method to transform a simple probabilistic distribution into a more complex one using inevitable and differentiable functions. They can be used as generative models by sampling from the base distribution and applying the the mapping function to the more complex distribution [51]. As generative models, they have seen success in tasks such as audio [46] and video [52] synthesis.

A visual summary of the GMLMs discussed in this section is provided in Figure 4, showing both the main algorithms discussed, as well the variants that are developed from them.

4 SYSTEMATIC MAPPING STUDY

We now turn attention to the first purpose of this article, the mapping study, where our goal is to understand how GMLMs have been used in intrusion detection in the published literature. This section describes our methodology and presents the main results of the study. Additional results of the mapping study focused on *evaluation metrics* are presented in Section 5. Addressing the

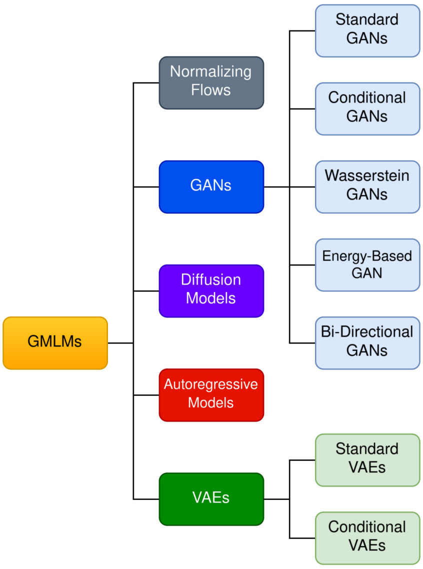


Fig. 4. Summary of GMLMs and associated variants reviewed in Section 3.

second purpose of this article, the works included in the mapping study are reviewed in detail in Sections 6 through 8.

4.1 Methodology

To select papers for the mapping study, two methods have been applied. The first is a series of web searches that were conducted starting in May 2020. These searches were made using Google Scholar, with the terms "generative adversarial network," "variational autoencoder," or some other generative model, paired with terms that would be related to intrusion detection, such as "intrusion detection," "malware detection,"and "cyber attack data." From the papers selected using these searches, we then used snowballing to find additional papers that met our selection criteria.

Using either method, the criteria for selection was the same. Two questions were asked for each paper evaluated: (1) Does the paper use a GMLM to solve a problem? and (2) Is that problem related to intrusion detection? For the second criterion, relation to intrusion detection did not necessarily require that intrusion detection be mentioned in the paper, so long as the problem was applicable. For instance, one issue of consideration was that of IDS dataset availability. A paper may use GMLMs to produce synthetic cyber security data that could be used by an IDS, but not refer to it explicitly as an IDS dataset. This relation to intrusion detection is not extended, however, to works in adjacent topics that lack a security-related context. For instance, anomaly detection is a frequently used technique for implementing IDSs, but a work such as Skip-GANomaly [5] that is evaluated on the CIFAR-10 image dataset would not be considered relevant to intrusion detection. However, derivative works which adapted it to intrusion detection tasks would be considered.

A paper that met the selection criteria would still be excluded if it was not written in English. Additionally, pre-print papers were excluded unless they were particularly noteworthy. A pre-print that had at least 50 citations would be considered influential and well reviewed enough for inclusion in the study.

A total of 53 papers that had been reviewed were found to meet the selection criteria. From these, to perform a more effective analysis, it was necessary to find a way to effectively categorize them.

Table 1. Overview of Papers Selected for the Mapping Study, Categorized by Topic Area

GMLM in IDS Application Topics	Discussed in	List of Papers
1. False Data Attacks	Section 6.2	[3, 41, 68, 81, 90, 108]
2. Evasion Attacks	Section 6.3	[36, 43, 58, 105, 109]
3. Exploratory Attacks	Section 6.4	[93]
4. Dealing with Unbalanced Datasets	Section 7.2	[26, 39, 44, 57, 60, 67, 87, 89]
5. Cyber Alert Generation	Section 7.3	[102–104]
6. Flow Generation	Section 7.4	[63, 82, 88, 114, 120, 123, 129]
7. Standalone GMLM IDSs	Section 8.2	[6, 25, 30, 38, 40, 42, 61, 70, 72, 75, 99, 115, 116, 126]
8. Hybrid GMLM IDSs	Section 8.3	[18, 21, 27, 47, 48, 107, 117, 119]
9. Limitations on GMLM IDSs	Section 8.4	[122]

To do this, two research questions were asked concerning the role of the GMLMs with respect to IDSs:

RQ1: How do GMLMs improve IDS testing?
RQ2: How do GMLMs improve IDS training?

4.2 Mapping Study Results

An analysis of each paper's main topic finds three broad categories that answer RQ1 and RQ2. For RQ1, we find works where GMLMs are used for penetration testing purposes. For RQ2, we find that training is often improved by GMLMs by addressing issues related to IDS datasets. In answering both of these questions, we also discovered a third category that is related to both testing and training, which is the use of GMLMs as IDSs. Figure 1 shows the relationship between each category and the two research questions, along with a breakdown of each category into several topics to be discussed in Sections 6 through 8. Table 1 provides a list of the papers selected, catalogued by application area topic. Further analysis of the papers is provided in Sections 6 through 8.

As each paper selected necessarily makes use of GMLMs, we have also created a mapping showing the number of times different GMLM types are used in each application area that we have explored. For GMLM types, we considered nine categories:

- four variants of GANs,
- three variants of VAEs,
- hybrid GMLM, and
- a category we named other GMLM.

This data is presented in the bubble chart in Figure 5.

When considering all application areas combined, we find the Standard GAN algorithm to be the most common GMLM type represented. However, where application areas are considered individually, it is only the most common GMLM type where penetration testing is concerned. The other two application areas are much more diverse in terms of GMLM types, and in particular, many GMLMs that were being used as IDSs themselves were found to use some form of hybrid architecture that either used multiple GMLMs or a GMLM paired with some other machine learning model. These works are discussed in Section 8.3 on "Hybrid GMLM IDSs."

5 ANALYSIS OF EVALUATION METRICS

In each of the areas where GMLMs have been applied to IDS applications, a different set of evaluation metrics has been employed. Figures 6 and 8 provide data on the number of occurrences

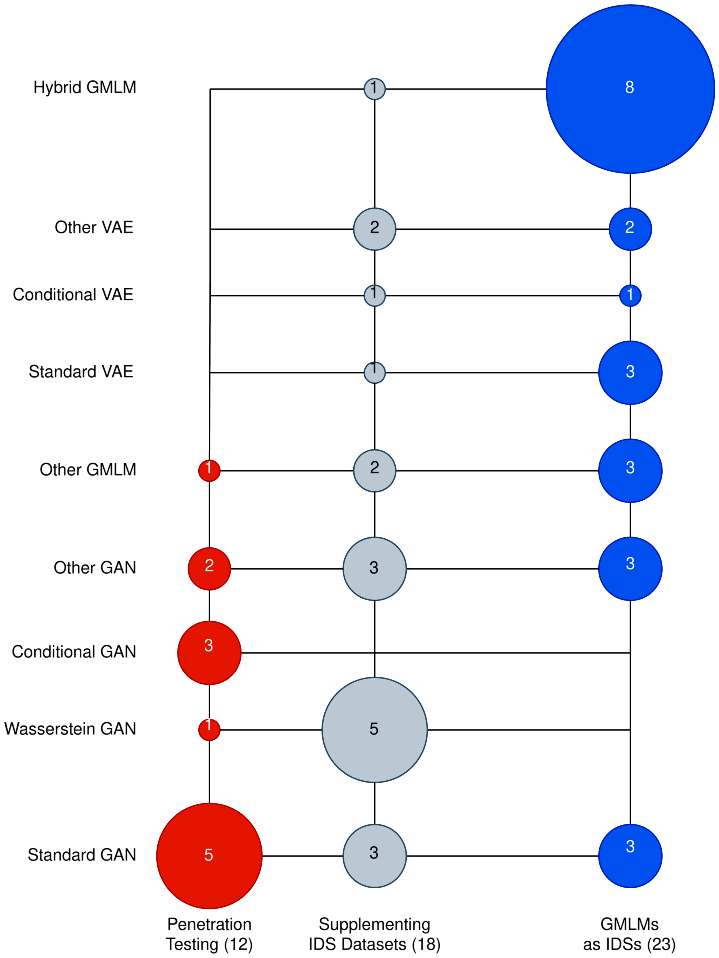


Fig. 5. Bubble chart showing number of times each type of GMLM was used in each application area for the 53 papers included in the mapping study.

of different types of metrics in each of the three application area for the works explored in this survey. Further analysis of the metrics discovered in these works and review of the underlying metrics notions is provided in this section.

5.1 Metrics Used for Evaluating Classifier Performance

A common group of metrics used in all application areas are those that can be used for evaluating classifier performance. In the GMLMs as IDSs application area, we find that these metrics are used almost exclusively, since the generative models here are being used in their capacity as classifiers rather than purely to create synthetic data. A breakdown of their frequency in this domain is provided in Figure 6.

These classifier metrics can be defined in terms of four quantities, obtained from the classifier's predictions on the test set. These are the true positives (*TP*), false positives (*FP*), true negatives (*TN*), and false negatives (*FN*). In the context of intrusion detection, a positive refers to the detection

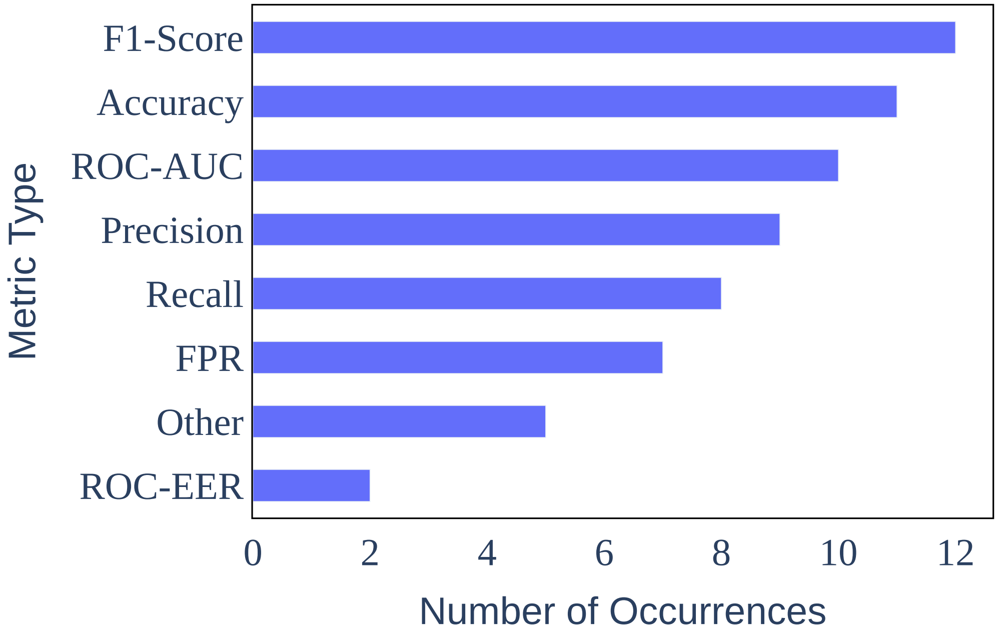


Fig. 6. Counts of evaluation metrics used in the papers reviewed in the GMLMs as IDSs application area. Note that some papers may use more than one metric, so the sum of the counts may be larger than the number of papers.

of attack data, whereas a negative refers to the detection of benign data. Based on these counts, Equations (2) through (6) provide definitions for each of the classifier metrics discovered in this application area.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

$$\text{Recall} = \text{True-Positive Rate (TPR)} = \frac{TP}{TP + FN} \tag{4}$$

$$F_1\text{-Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 * TP}{2 * TP + FP + FN} \tag{5}$$

$$\text{False-Positive Rate (FPR)} = \frac{FP}{FP + TN} \tag{6}$$

In addition to these metrics, it is common for classifiers to be evaluated in terms of a **Receiver Operator Characteristic (ROC)** metric. This is not a single value but a curve on a graph that considers several different thresholds of classification performance. The x-axis of this graph considers the false-positive rate of the classifer at each threshold, and the y-axis shows the corresponding true-positive rate. From this curve, two metrics are derived for classifier performance. The first is the **Area under the Curve (AUC)**. The higher this value is, the less false positives are necessary for true positives to be classified correctly. Intuitively, if this value were equal to 1, then the graph would resemble a horizontal line from $y = 1$, and perfect classification would be possible. The second metric derived from ROC is the **Equal Error Rate (EER)**, which is the point on the graph

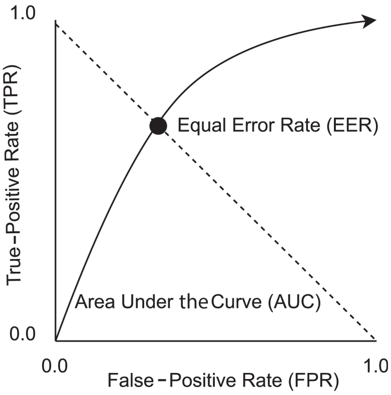
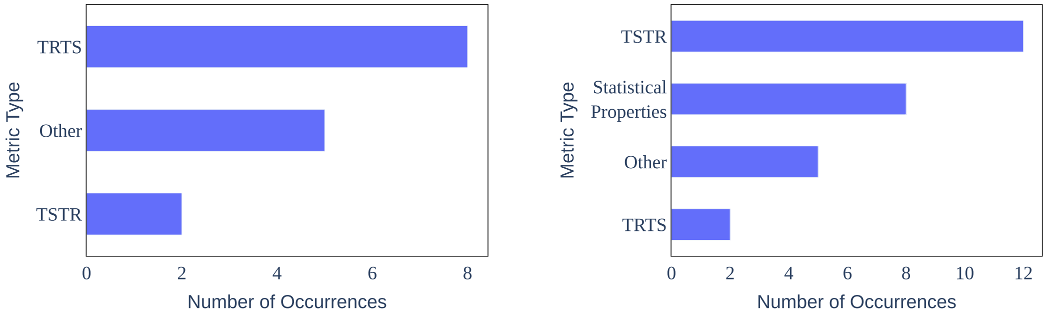


Fig. 7. Example of a ROC curve plot, showing the AUC and the location of the equal error rate.



(a) Penetration testing

(b) Supplementing datasets

Fig. 8. Counts of evaluation metrics used in the papers reviewed in the penetration testing/supplementing datasets application areas. Note that some papers may use more than one metric, so the sum of the counts may be larger than the number of papers.

where the false-positive and false-negative rates are equal. Figure 7 provides an example of a ROC curve along with the location of the EER.

5.2 Metrics Relative to Testing and Training

Although classifier metrics are common in all application areas, what is not common is which classifiers are being evaluated and how those classifiers are trained. In the penetration testing and supplementing IDS datasets application areas, two approaches are noted. The first is the case where a classifier is *trained on real data and tested on synthetic data*, which we denote with the abbreviation TRTS. The second is the reverse scenario, where data is *trained on synthetic data and tested on real*, denoted by TSTR.

We note in Figure 8(a) that the TRTS metrics appear much more commonly in penetration testing applications. Here, the scenario is often that one is presented with an existing classifier being used for intrusion detection, and we wish to worsen its performance by constructing synthetic attacks. Therefore, synthetic data should be the test in the scenario rather than the training data. In contrast, TSTR is used in the penetration testing domain only in a minority of cases, and with the goal of determining whether retraining the original IDS with the adversarial data can improve performance.

5.3 Metrics in Supplementing IDS Datasets

For applications where the goal is to supplement IDS datasets and improve performance, a wider variety of metrics are employed. In Figure 8(b), we find that two classes of metrics dominate. The first is TSTR, which we use to evaluate the quality of synthetic data by how well it improves the performance of the classifier. Since the primary goal of these applications is to improve IDS performance in the first place, this choice of metric is self-explanatory. The second highest group, however, is labeled as statistical methods. These are methods that measure differences between the statistical distributions of the real and synthetic datasets. Some of the metrics in this group include the following:

(1) *Jensen-Shannon Divergence*, which has been referred to as "the increment of Shannon's entropy"and is used to compute the difference between random graphs [66].
(2) *Histogram Intersection* is a method originally developed for comparing the similarity of images [101]. For each image, a histogram is constructed from the frequency of each color. The minimum frequency for each color is then taken in the corresponding intersection.
(3) *Conditional Entropy* is the total amount of information given in some variable X given that another variable Y is some particular value [20].

The goal with statistical methods for evaluating synthetic data is not necessarily concerned directly with improving classifier performance but rather to ensure that the data is realistic in the first place. The purpose of training with synthetic data is to harden an IDS against real attacks, which we do not have data for. Evaluating purely in terms of TSTR may demonstrate that we can improve classifier performance, but it could be against the wrong attacks. To this end, statistical methods should be considered at least as an auxiliary metric for GMLMs used for supplementing datasets.

Other metrics which focus more on data realism include TRTS and metrics based on domain knowledge. For the domain knowledge checks, the exact qualities being checked for will be specific to the data generated, and there is no standard method adopted for a specific kind of IDS data. Further work is needed to determine an ideal method of determining data realism.

5.4 Other Relevant Metrics

Not every metric that has been used in GMLM evaluation is currently seeing common use with GMLMs that are used for IDS applications. The following are some metrics worth considering in future research:

(1) **Fréchet Inception Distance (FID)** is a metric that measures the distance between distributions using their mean and covariance. Where GMLMs are concerned, this is typically used to measure the distance between real and synthetic distributions. FID has established itself strongly in research using GANs, for applications such as generating medical data [12] and image generation [37]. Its absence within the surveyed works is therefore somewhat surprising, particularly given the number of works which use GANs specifically for addressing IDS-related problems.
(2) **Real to Real (RTR)** and **Synthetic to Synthetic (STS)** Similarity are a pair of metrics that concern the distributions of real and synthetic data. Each employs the average cosine similarity of elements within a distribution to other elements within the same distribution [73]. When STS and RTR similarity are close in value, it implies that the real and synthetic data belong to similar distributions.

6 GMLMS FOR ASSISTING WITH PENETRATION TESTING OF IDSS

6.1 Background

A component of good cyber security is testing to find the vulnerabilities in one's own system. This is known as *penetration testing* (or pentesting), and it is a common practice in industry. Pentesting is usually performed manually; however, efforts have been made to automate the process and realistically simulate a human hacker [35]. Existing work shows that GMLMs are useful tools in performing this task. Among the literature where GMLMs are used for pentesting and similar applications, three types of simulated attacks show prominence. These are *false data attacks*, *evasion attacks*, and *exploratory attacks* (Figure 9). Sections 6.2 through 6.4 detail several works for each attack type where a GMLM is constructed to apply that attack for testing cyber defenses. Some of the works discussed in these sections may be applicable to multiple application areas, such as by demonstrating the use of a false data attack to evade defenses, although they are only discussed in one section. Section 6.5 synthesizes the works surveyed in the section and points out directions for further work.

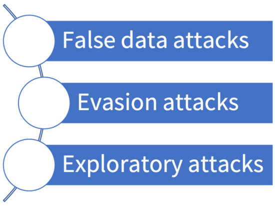


Fig. 9. Section overview.

6.2 False Data Attacks

Rigaki and Garcia [81] seek to simulate human behavior in their own work which leverages a GAN to evade the network-blocking functionality of the Stratosphere **Intrusion Prevention System (IPS)**. Where automated pentesting would typically simulate a hacker, however, they instead simulate Facebook chat messages to make communication over a Command and Control (C2) channel appear legitimate. The authors configured Stratosphere to model normal network traffic and block all IP addresses whose communication differed from what was standard. They trained a GAN against the blocking behavior of the IPS by checking what percentage of traffic was blocked, unblocked, or having no action taken. After around 300 to 400 training epochs, the traffic would never be blocked.

IDSs and IPSs rely on machine learning to distinguish between malicious and benign behavior, but GANs are designed through their adversarial process to force their generated data to be misclassified. The work of Rigaki and Garcia [81] demonstrates how GANs may be used to circumvent IPS defenses, which may prove useful in automated pentesting. While most realistic scenarios would require an IPS to accept a more broad spectrum of network traffic than just Facebook chats, there are still some networks which have similar limitations. One such example is in **Supervisory Control and Data Acquisition (SCADA)** networks, which are more well behaved due to using a limited number of protocols and communicating in a mostly predictable manner [10].

Ahmadian et al. [3] explore the use of GANs for simulated attacks on SCADA networks, using **False Data Injection (FDI)** to manipulate the price of electricity in the simulation. In this scenario, an attacker is able to perform a Man in the Middle (MITM) attack on a remote terminal unit that is responsible for providing measurement information to the SCADA network. The attacker wishes to create fake congestion on this network so that the price of electricity would rise, allowing the attacker to sell electricity purchased on a previous day at a higher price. The GAN is used to generate measurement information that would be perceived as real by the power system state estimator.

This type of attack has some potential to be realistic, as FDI attacks are able to be carried out stealthily against SCADA networks, provided that the attacker has knowledge of the system model [76]. In their simulation, Ahmadian et al. [3] use measurement information from public access websites, which could be used as an aid to produce realistic data. Nevertheless, the requirement on having access to a certain level of knowledge makes this use of GANs not quite bulletproof. An energy system operator could deliberately publish fake information to isolate a remote terminal unit suspected of reporting false data, which would poison the GAN to produce much less stealthy FDI attacks.

Several other works explore the use of GMLMs for conducting FDI attacks against SCADA or other cyber-physical networks. Most of these works use GANs over other GMLMs. For instance, Jiao et al. [41] use a GAN with a self-attention module for conducting FDI against the IEEE 14-bus test system. Mohammadpourfard et al. [68] and Shahriar et al. [90] both use CGANs for the same task, with the work of Shahriar et al. using a conditional WGAN. This task does not appear to require significant changes to the original GAN algorithm, however, as Tong and Qi [108] demonstrate using a standard GAN.

6.3 Evasion Attacks

Lin et al. [58] propose the use of a GAN to manipulate malicious network traffic to evade IDS defenses. To avoid negating the traffic's attack functionality, the authors consider different categories of features for network traffic (Intrinsic, Content, Time Based, Host Based), and whether they are considered functional features (i.e., whether the functionality of the attack would change if they were changed) for different categories of malware. They then train a GAN on the NSL-KDD dataset that will generate malicious examples where the functional features are unmodified and test against a black-box IDS for different categories of malware. Although this work achieves good performance in fooling the black-box IDS, it raises some questions for future work. First, there are some concerns for real-world scenarios because of the need to determine non-functional features for each attack type. Second, the experiment could be repeated to explore host-based cyber attacks and what features are considered non-functional in those contexts.

Hu and Tan [36] address the latter of these questions. Their MalGAN framework is aimed at convincing a black-box IDS that malicious executables are actually benign. The functional features the MalGAN authors are primarily concerned with are binary features—due to their high contribution to detection accuracy—and more specifically they focus on Windows API features. Furthermore, rather than removing API calls, new API features are added. Using MalGAN, the authors are able to achieve true-positive detection rates of no more than 0.2% when the IDS is made to classify their adversarial samples.

Kawai et al. [43] attempt to improve upon the methodology of MalGAN. The authors consider a number of problems with the original paper, most notably that the number of features from the original malware samples are reduced. They also consider that only one malware should be used, as using multiple types might impact performance, and the likely use case is to generate a variant of just one type. Accordingly, the authors' improved MalGAN adds in API features from cleanware to malware, and found that this increases the evasion rate. They conclude that a detector could decrease its ability to be evaded by decreasing the number of API features, but this would also increase the false-positive rate.

Both IDSGAN [58] and MalGAN [36] expose a vulnerability in IDSs to adversarial perturbations. Given their ability to evade current defenses, GANs may become a standard component of the future attacker's toolkit. Accordingly, pentesters will also need to make use of them to ensure that systems are hardened against GAN-based attacks.

This naturally supposes a question, however—What is the best defense against a GAN-based attack? To combat this type of attack, Usama et al. [109] propose the use of a GAN to produce adversarial examples to retrain an IDS. In their paper, the authors compare the performance of eight different machine learning algorithms for classifying attacks before and after GAN-based adversarial perturbations were introduced, as well as after GAN-based adversarial training was employed to counter the GAN-based attacks. The results showed that while GAN-based attacks significantly reduced classification performance of the IDS, GAN-based adversarial training reduced this drop in performance to only a few percentage points.

Aside from GANs, **Large Language Models (LLMs)** may also be used in conducting future cyber attacks. Tann et al. [105] compare three publicly available LLMs (ChatGPT, Google Bard, and Microsoft Bing) in their ability to answer various cyber security related questions. Among these were questions related to Capture-The-Flag challenges that would require performing exploits such as shell shock attacks or buffer overflows. While not every question was able to be answered, in part due to safety standards implemented to prevent this type of use, several Capture-The-Flag challenges were nonetheless successfully completed, with ChatGPT completing a majority of challenges.

6.4 Exploratory Attacks

In addition to evading an IDS, a GMLM may be used to perform an exploratory attack and learn to mimic it. Shi et al. [93] consider the case of a black-box API for an application that performs classification tasks. The number of API calls that it allows in a given time period is limited, and the authors further consider that making the maximum amount of API calls in each period would be considered malicious and result in blocking. Accordingly, the authors use a GAN to reduce the amount of API calls needed by adding in synthetic data that mimics what they have already tested.

With the newly learned classifier, Shi et al. [93] consider that further attacks can be launched. Aside from simple evasion, an attacker can also launch a causative attack, where adversarial training data is provided to the black-box classifier that would cause its performance to decrease. This is also known as a *data poisoning* attack and has several documented defenses against [97].

6.5 Discussion

We synthesize the works reviewed in Sections 6.2 through 6.4 and identify a few cross-cutting themes and directions for future research.

Exposing Vulnerabilities. The works discussed in this section demonstrate that GMLMs can be effective in exposing the weaknesses in both host- and network-based IDSs. Thus far, GMLMs have been found most effective in performing FDI attacks. These attacks pose a threat to critical infrastructure domains such as smart grids, finance, and healthcare. Presently, although countermeasures have been developed for FDI, it has been identified that there is a lack of suitable datasets for evaluating IDSs against them [4]. Since the purpose of GMLMs is to produce realistic but otherwise synthetic data, their applicability to this task appears obvious. Future work toward creating publicly available FDI datasets using GMLMs should be strongly considered.

Malware Detection. GMLMs have been explored to a lesser extent for other types of attacks, such as evading malware detection, although they already show some promise, at least when evaluated in terms of IDS performance. Just because an IDS no longer detects a piece of malware, however, does not mean that that this malware can be used to infect a system. A proof of concept should be considered in future work to demonstrate a cyber attack that

(1) without modification is detected successfully by an IDS,

Table 2. Highlights of Major Success Stories and Opportunities for Improvement in the Works of Each Attack Type Reviewed in Section 6

	Major Successes	**Opportunities for Improvement**
False Data Attacks	Demonstration of attacks in simulated SCADA environments	Explore use of GMLMs other than GANs
Evasion Attacks	Successful evasion of malware detections	Conduct more practical tests
Exploratory Attacks	Reverse engineering of black box environments	Topic is under-explored

(2) uses adversarial samples generated by a tool such as IDSGAN [58] or MalGAN [36] to successfully evades IDS defenses, and
(3) achieves the same results when the attack is performed with the adversarial samples as are achieved when the original attack is performed.

Such a proof of concept would demonstrate much more clearly whether GMLMs are useful in malware detection.

Offensive Utility. One concern that might be raised about GMLMs being used for penetration testing is whether they may be used by attackers in the future. If the offensive utility in GMLMs is much greater than the defensive utility, this would suggest that there are ethical concerns for continued research of GMLMs in this domain. However, given that proof of concept attacks have been demonstrated using GMLMs, it is possible that state actors interested in using GMLMs for use in cyber weapons are researching these capabilities. Fortunately, an existing work [109] has explored the use of GMLMs in defending against GMLM-based attacks. Future work should expand on this and demonstrate how IDS performance can be improved upon whenever GMLMs are used to decrease performance as part of penetration testing.

Going beyond GANs. Additional future work in penetration testing should consider the use of GMLMs other than GANs. Thus far, penetration testing appears to be the least researched application area for IDSs of the three explored in this article. This is also reflected in the diversity of works. Many of the works discussed in this section do not depend specifically on the architecture of GANs and therefore could be replicated with other GMLMs such as VAEs or autoregressive models to see how much more effective they would be at bypassing IDSs, or improving their performance against such adversarial training.

We conclude our discussion of the works around penetration testing reviewed in this section with an overview of some of the major success stories and opportunities for future work, which is provided in Table 2.

7 GMLMS FOR SUPPLEMENTING IDS DATASETS

7.1 Background

One of the greatest problems facing intrusion detection research is the availability of quality labeled data. Many existing datasets are upward of a decade old or more [92] and thus cannot reflect the current state of malware, which is constantly evolving. There are a number of reasons for this situation. Producing labeled security data requires expert knowledge. The task can be extremely laborious and is nevertheless not guaranteed to be correct. There are also concerns with privacy and classified data that may require data to be anonymized, which is also an imperfect process and prone to human error [65].

There exist some works in the literature that address the need for generated data but do not involve the use of GMLMs. In 2008, Brauckhoff et al. [14] created FLAME, a tool that can inject anomalies into a dataset of normal background traffic, although the anomalies still have to be user defined. More recently, Applebaum et al. [8] created CALDERA, which allows for automated simulations of attacks on a network. This could in turn be used in combination with network monitoring tools to generate a dataset, although human effort would still be needed to add labels.

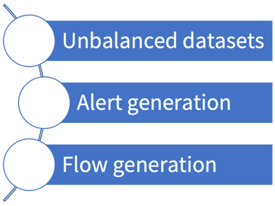


Fig. 10. Section overview.

Thus far, these tools have yet to yield significant impact on public datasets. Other tools for generating security data should therefore be explored. This section details how GMLMs, and especially GANs, are applied to this task supplementing IDS datasets (Figure 10). Section 7.2 concerns unbalanced datasets, which is a problem that a significant amount of GMLM research is conducted to address. Sections 7.3 and 7.4 show how GMLMs are applied to two specific types of IDS data generation tasks: cyber alert generation and flow generation. Finally, Section 7.5 synthesizes the works surveyed in Sections 7.2 through 7.4 and discusses the strengths and limitations of the approaches.

7.2 Dealing with Unbalanced Datasets

In some cases, while the datasets used to train IDSs may be sufficient in size, they may contain one or more classes that are dwarfed by the rest. Fiore et al. [26] attempt to address this problem of unbalanced datasets in the context of credit card fraud detection, where there is an under-representation of fraudulent data in existing public datasets. GANs are introduced here as a method of over-sampling the minority classes. The authors evaluated a classifier trained on the base dataset compared to a dataset augmented with synthetic data. They found that sensitivity can be increased with such an approach, albeit at a cost of increasing the false-positive rate.

Merino et al. [67] also attempt to address the same problem within a more traditional IDS dataset. Their work generates new instances of the Neptune, Smurf, and IP Sweep attacks in the KDD99 dataset, finding that 100% of the generated attacks were classified correctly. These were, however, not the most under-represented attacks in the dataset, with several attacks having less than 10 instances in the dataset and the most under-represented having just 2.

Shahriar et al. [89] provide a framework using a GAN to improve the NSL-KDD99 dataset with synthetic samples. Their work demonstrates consistent improvement when training an IDS on the new hybrid dataset (consisting of both synthetic and real data) compared to the original dataset. This performance is measured in terms of precision, recall, and F1-Score, and is provided independently for each class label for each model of S-IDS (the Standalone IDS) and G-IDS (the IDS trained on GAN data).

Yilmaz et al. [121] provide a similar solution of creating a hybrid real/synthetic dataset for addressing balance issues in the UGR'16 dataset. This dataset is divided into several sub-datasets with between 22 and 71 attack samples, and thousands of non-attack samples. In each case, the authors generate a number of attack samples to match the non-attack samples (making the new dataset 50% attack and 50% benign), and evaluate a classifier across the original and hybrid datasets. When evaluated in terms of precision, recall, and F1-Score, the classifier shows significant improvement over the hybrid dataset compared to the original.

A unique solution to the balanced dataset problem is presented for host-based IDS by Salem et al. [87]. Their strategy is to use a Cycle GAN, a type of GAN that can use data from one distribution and map it to another. In this instance, they transform benign behavior in the ADFA-LA dataset into anomalous behavior. The results were mixed, however. Compared to an IDS trained on the unbalanced data, an improvement was seen in both recall and the F1-Score, but the number of false positives increased from 31 to 169.

Some non-GAN solutions to the unbalanced dataset problem have also been explored. For instance, Lin et al. [57] use a VAE to generate new abnormalities in several IDS datasets and achieve an improved F1-Score over the originals when evaluated using a multi-layer perceptron. Khanam et al. [44] use a similar approach, generating new samples for NSL-KDD with a VAE and a custom loss function and evaluating with a deep neural network. Lopez-Martin et al. [60] further expand upon the use of VAEs for the unbalanced dataset problem by allowing the VAE to accept labels as an input to either the encoder or the decoder. One notable use of a model that is neither a VAE nor GAN for fixing unbalanced datasets was the use of Normalizing Flows by Idrissi et al. [39] to produce pseudo-attack samples for an anomaly-based network IDS.

7.3 Cyber Alert Generation

Sweet [102] proposes the use of a GAN for generating alert data that mimics the output of a **Network Intrusion Detection System (NIDS)** in response to an attack. These alerts provide a summary of the network traffic features that the NIDS believes to be malicious, such as IP addresses, ports, payloads, and timestamps. They also provide an alert category indicating what type of attack the suspicious data could be classified as. Sweet notes that because cyber alert data is only created after an attack has already taken place, it is currently only used for reactionary defense techniques, but that there may be a use for this type of data in proactive cyber security.

Sweet develops two WGAN (discussed in Section 3) models to generate this alert data [102]. The first model uses a gradient penalty (WGAN-GP), and the second extends it with a mutual information constraint (WGAN-GPMI). The same two models are used in further research by Sweet et al. [103, 104] with respect to alert generation. Acknowledging that there is not yet a standard metric for analyzing the fidelity of generated data, Histogram Intersection is used as an approach for such a purpose in all of the works [102, 103]. In later work, this same model is re-evaluated using Jensen-Shannon Divergence, Conditional Entropy, and Joint Entropy [104].

In the conclusion to the work of Sweet [102], it is written that while GANs have promise in generating NIDS alerts, the results are far from perfect for several reasons. Among the reasons are the lack of malicious alert data for training GANs in the first place, the need to be able to generate sequential alert data, and the need for general improvement in quality. The first of these may be the most difficult hurdle to overcome, as it suggests that an investment in human labor may be needed to resolve a bootstrapping problem with GANs—that they require a significant amount of data to perform the role of generating more.

7.4 Flow Generation

Ring et al. [82] explore the idea of generating datasets of sufficient size and variance to improve anomaly-based intrusion detection. Focusing on network flow data in particular, their work uses an improved version of WGAN to expand upon netflow records in the publicly available CIDDS-001 intrusion detection dataset [83].

Ring et al. [82] address several problems with GANs for this task. The first is that flow data often contains categorical features, such as IP addresses and ports, whereas GANs are typically suited to working with continuous features. The authors consider several solutions to this. Categorical data could be converted into one-hot representation, although this is less viable when the number of IP addresses and ports in the dataset is particularly large. Representing the data as individual binary features based on their integer representations (i.e., a port would become 16 separate features) does not suffer from this problem. The authors also develop a third solution based on a tool called *IP2Vec*, which can generate a vector encoding that takes contextual information from the dataset into account.

An alternative solution to encoding netflows for GAN generation is proposed by Manocchio et al. [63]. In their work, it is noted that IP2Vec may produce particularly large feature vectors when there are many IP addresses in the dataset. Instead, the authors propose one-hot encoding the digits of the base-10 representation of IP addresses. The IP address is split into 12 digits, each of which is represented by a one-hot vector for each possible value from 0 to 9. The first digit of each octet is represented by only the values 0-2. As the authors note, this does have a small possibility of generating invalid addresses such as 266.277.288.299, yet the GAN should learn to avoid such possibilities. For ports, a similar encoding structure can be used, but it would necessarily have the same issue, as a port of 67000 might possibly be generated.

In addition to issues related to categorical features, netflows are inherently temporal. The ability for a flow to be considered as an anomaly may be dependent on other flows in the same dataset. To address this issue, Yin et al. [123] use time-series GANs and demonstrate higher accuracy compared to several other generative approaches.

Aside from being difficult to represent, the features present in netflow data may not be sufficient to generate realistic network traffic. Shahid et al. [88] note that most works for synthetically generating network traffic focus either on purely flow-level or packet-level features. In their own approach, they generate sequences of packet sizes (a packet-level feature) with an autoencoder and use this to inform flow generation with a GAN.

Another issue discussed by Ring et al. [82] is that there does not exist a single, widely accepted methodology for evaluating generated network data. For this problem, the authors use data visualization tools along with Euclidean distance metrics to evaluate the diversity and distribution of the data while relying on domain knowledge to evaluate the quality of the data. This domain knowledge consists of seven heuristics used as sanity checks. The sanity checks test for undesired behaviors, such as the presence of TCP flags in UDP traffic, or the use of UDP for normal user behavior that typically occurs over TCP, such as HTTP traffic.

While these are all perfectly reasonable sanity checks for generated network traffic, the number of ways in which traffic may differ from realistic data is certainly larger than the seven heuristics proposed by the authors. Relying on human knowledge can be particularly time consuming, so an alternative may be to simply ask how the synthetic data will perform when used by an intrusion detector. For this, Zingo and Novocin [129] propose the “GAN vs Real”metric, which compares the accuracy of a classifier trained on the synthetic data and tested on the real data, to the same classifier trained and tested on real data. This can be considered as an extension of the TSTR metric introduced in Section 5.

GANs are not the only generative models applied to netflow generation. Some work has been done to apply VAEs to the same task. Yang et al. [120] use conditional VAEs to generate new anomalies for the NSL-KDD [106] and UNSW-NB15 datasets [69], and evaluate them with a deep neural network using metrics such as accuracy and false-positive rates. Additionally, Xu et al. [114] provide a unique approach using autoregressive neural networks for generating network traffic, which is compared against GAN approaches as well as Bayesian networks and Gaussian mixture models.

7.5 Discussion

We synthesize the works reviewed in Sections 7.2 through 7.4, and identify a few cross-cutting themes and directions for future research.

Table 3. Highlights of Major Success Stories and Opportunities for Improvement in the Works of Each Topic Reviewed in Section 7

	Major Successes	**Opportunities for Improvement**
Dealing with Unbalanced Datasets	Improved balance in datasets; increases in IDS performance	Need for data to bootstrap experiments
Cyber Alert Generation	Creation of realistic cyber alert data; demonstration of statistical methods for evaluating synthetic data quality	Need for data to bootstrap experiments; better methods for generating sequential alert data
Flow Generation	Creation of realistic netflow data	Need for standards for metrics and feature extraction

Improving Balance in Datasets. Within the works discussed in this section, one can find myriad GMLM architectures that have been successfully applied to the task of improving balance in intrusion detection datasets. There are limitations to this application, however. To generate data that fits some distribution, there must be data to form that distribution in the first place. Otherwise, the best a GMLM could do to find statistically similar samples to the minority class would be to copy existing samples verbatim. The exact threshold for the number of samples necessary for GMLMs to show an improvement in the dataset has not been studied sufficiently. Anecdotally, however, we have seen works create improvement in minority classes with as few as 22 samples [121].

Feature Extraction. The type of security data being generated can impose further difficulties when attempting to solve balance-related issues. In particular, research into netflow generation has shown that feature extraction can be a significant hurdle for synthetic generation. Features in security datasets are often highly categorical and thus will need to be represented as several features when encoded into a feature vector. Each of the methods of encoding in this way comes with tradeoffs, however. The number of features generated should not be too high (as would be seen with one-hot encoding), but it is also important to not use an encoding that allows for the generator to produce invalid values (e.g., the IP address 266.277.288.299).

Variable Data Length. Generating other types of security data, such as malware samples or malicious packet captures, may experience greater issues with regard to feature extraction. Where netflow data is merely highly categorical, executables and packets both contain sequences of variable length of data that may or may not be highly categorical, depending on context. All of this needs to be transformed into fixed-length feature vector.

The work of Shahid et al. [88] touched on the issue of variable length with respect to network packets by padding the end of the packets with zeroes. This allows it to at least be transformed into a feature vector, but the individual bytes were still encoded as plain integers. If a different network protocol were used, there would likely need to be a different mapping to account for bytes that represent categorical data.

This strategy of simply padding bytes would not be applicable when generating samples for malware classification. Indeed, encoding byte sequences in malware is considered unreliable; however, there are a diversity of other feature extraction methods with varying levels of utility [1]. These include binary features for the presence of various strings or API calls, function level features, and encoding the program as a control flow graph. The degree to which any of these features can be effectively generated by a GMLM should be explored in greater depth in future work.

We conclude our discussion of the works around supplementing IDS datasets reviewed in this section with an overview of the major success stories and opportunities for improvement under each topic; see Table 3.

8 GMLMS AS IDSS

8.1 Background

Rather than functioning as a supporting tool, a GMLM can alternatively perform the role of the IDS itself. The way this task is performed depends upon the type of GMLM being employed. For VAEs, a common approach is to use reconstruction techniques. The model learns what benign traffic should resemble and uses reconstruction loss to determine whether a new sample is considered an anomaly. With GANs, the discriminator is often as a multi-class classifier that determines if a sample represents an intrusion, benign activity, or if it is fake. The generator for the GAN IDS typically plays the same role as it does when the IDS is separate from the GAN—increasing the number of samples from minority classes and hardening the classifier against novel attacks.

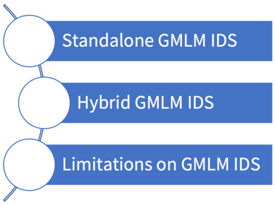


Fig. 11. Section overview.

These novel attacks in particular are some of the more important cyber attacks for GMLMs to be employed against. These can come in several forms. The first are zero-day attacks [100], which describe attacks that exploit undisclosed vulnerabilities. Additionally, there is metamorphic malware [86], which creates variations of itself with different instructions that have the same effect, and polymorphic malware, which encrypts its malicious payload and uses metamorphic techniques on the code responsible for decrypting it. All of these types of attacks can pose a difficult problem for IDSs, which may not be trained on datasets that provide the level of variation necessary for dealing with unforeseeable threats. Accordingly, many of the papers which use GANs as IDSs directly are concerned with handling some form of novel cyber attacks.

The following subsections discuss different implementations of GMLM-based IDSs (Figure 11). Section 8.2 discusses works that use a singular GMLM for intrusion detection, whereas Section 8.3 relates to works which combine GMLM techniques as part of a larger hybrid IDS. Works selected for either of these sections must use one or more GMLMs for the task of classifying cyber attack data as either benign or malicious. Section 8.4 discuss some limitations of GMLM-based IDSs. Finally, Section 8.5 reflects on the works surveyed in the section and provides a taxonomy of how the models are evaluated relative to each other.

8.2 Standalone GMLM IDSs

The VAE as IDS architecture using reconstruction probability is originally provided in the work of An and Cho [6]. Zavrak and Iskefiyeli [126] provide a comparison using this algorithm of the capabilities of VAEs against traditional autoencoders, as well as a non-generative approach using support vector machines. Using the area under the ROC curve (AUC) they find a higher performance for VAEs for most anomaly types. Expansions on the VAE as IDS architecture include the work of Osada et al. [75] using semi-supervised training and the work of Sun et al. [99] on training VAE anomaly detectors on large datasets using sparse representation. In more recent work, Najari et al. [70] further demonstrate the capabilities of VAEs as well as Normalizing Flows in their work on generative robust anomaly detection. Their algorithm, which focuses on removing anomalies from benign data early in the training process, outperforms baseline models in scenarios with higher rates of anomalies.

LLMs may also show significant potential in their use for intrusion detection. Guastalla et al. [30] explore the use of GPT 3.5 and GPT 4 in their ability to detect DDoS attacks, training several configurations of each on a small sample size of network traffic. Aside from showing better performance than a baseline classifier model, the LLM was able to explain its reasoning for its classification decisions. The explanations, however, were less useful in more highly tuned configurations, which had better classification performance.

Ferdowsi and Saad [25] provide an excellent example of the GAN as IDS architecture. Although their model is distributed, with a discriminator placed on each device within an IoT network, the method of how the GAN is used for the intrusion detection task is the same. The generator produces anomalies, and the discriminators serve as both the IDS and as feedback for the generator. Their work compares several GAN-based IDS solutions for an IoT network, and evaluates the false-positive rates, accuracy, and precision for each one. On each of the metrics, their distributed solutions performed better than more centralized GAN solutions.

Yan et al. [116] show a similar interest in IoT intrusion detection and provide one of the first examples of diffusion models being used for IDS tasks. Other works exploring GMLM-based intrusion detection for IoT include those of Lopez-Martin et al. [61] and Xu et al. [115], which use conditional VAEs, as well as those of Idrissi et al. [38] and Nie et al. [72], which use GANs.

Kargaard et al. [42] consider the ability of GANs to combat intelligent malware, which could be designed to combat machine learning based defensive systems. Their work builds upon MalGAN [36] (which was discussed in Section 6), using the discriminator for malware classification. The authors use honeypots to capture real malware samples for training the GAN, which are then classified using the VirusTotal cloud service, and subject to feature extraction using a tool called *Cuckoo Sandbox*. To improve the stability of the discriminator, which now has to function as an IDS, the authors apply minibatching to the training process. It is unclear how effective the approach is since the paper lacks an evaluation of the GAN's performance.

Jan et al. [40] apply GANs to create an IDS for Android malware based on the application's behavior. To do this, they use a modified version of Android that captures the intents (data structures for communicating with the OS certain operations to be performed) of running applications. This allows them to generate a dataset containing the dynamic behavior of a set of benign applications. The authors then train a **Deep Convolutional Generative Adversarial Network (DCGAN)** on this dataset and use the discriminator as a means for detecting patterns of malicious behavior, which may include polymorphic malware. The role of the generator is to create variations on the input which would serve as the "malware."

Jan et al. [40] compare their DCGAN model to a number of other models published in a different paper [71] but tested on the same malware dataset. Although they achieve remarkably high performance figures, it is unclear whether this is because of the use of the GAN's discriminator as an anomaly detector or because dynamic features were used instead of static features. Regardless, the relatively low false-positive rate (0.002) is substantial and suggests that the approach could be promising in other environments.

8.3 Hybrid GMLM IDSs

The architecture of a GAN-based IDS is not limited to pure GANs. In the work of Kim et al. [47], the authors wish to use GANs for malware detection, with an emphasis on classifying zero-day attacks. However, they are concerned about the stability of GAN training and instead opt to first train an autoencoder model to produce a generator, then transfer that generator into a GAN so that it may be used to train a discriminator. This architecture is referred to by the authors as a tGAN (Transferred GAN).

The resulting model is tested on a dataset from the Kaggle Microsoft Malware Classification Challenge [85] and compared for accuracy against a number of other machine learning models. In one experiment, Kim et al. [47] reduce the amount of training data to between just 10 and 50 data points of each malware type, and show accuracy figures for each level compared to other models. Due to the relatively low amount of training data, the effect is that most of the testing dataset would be a novel attack type. The higher accuracy of tGAN compared to other models on this task reflects well on its ability to deal with these types of threats. One drawback here is that the authors did not report on false-positive rates, which are important to consider when evaluating the performance of a malware detection system.

In a follow-up work, Kim et al. [48], extend their work on tGAN by using deep autoencoders and provide additional statistical metrics for the new tDCGAN model in the form of precision, recall, and F1-Score. However, they do not provide the same metrics for other models to effectively evaluate the relative performance of tDCGAN.

Freitas de Araujo-Friho et al. [27] use GANs in a manner opposite to Kim et al. Rather than first training an autoencoder, and using the decoder as a GAN's generator, instead they first train a GAN and transfer the generator into autoencoder. The resulting autoencoder is then used for intrusion detection by using its reconstruction loss. The performance for this model is compared against two other GAN-based IDSs using an AUC metric, demonstrating that it has better classification performance compared to traditional GANs.

Hybrid models may also be trained without the use of transferring one model into another. In the work of Yang et al. [119], an "adversarial VAE" is trained using both a detector and a discriminator, with the generator and encoder being the same model from the beginning. After training, a softmax layer is appended to the generator/encoder to create a classifier for anomalies. Dinh et al. [21] also experiment with simultaneous training of generative models with a "Twin VAE" that combines a standard autoencoder with a VAE. In this model, the decoder of the VAE is matched with the encoder of the standard autoencoder.

Some hybrid models may also combine GMLM-based intrusion detectors with other, non-GMLMs. Yang et al. [117] provide an example combining clustering with a VAE. The VAE uses reconstruction error to distinguish between known and unknown attacks, whereas the cluster model learns the distribution of benign traffic. Other works exploring GMLMs in supportive roles with traditional classifiers include the work of Taylor and Eleyan [107], which uses conditional VAEs for dimensionality reduction when classifying malware, and the work of Dao et al. [18], which uses VAEs to provide an attention mechanism for a convolutional neural network based classifier.

8.4 Limitations on GMLM IDSs

The works discussed in the previous two subsections highlight that GMLMs can be effective for detecting attacks, but there are also limitations on this ability, particularly with regard to how useful the synthetic data is in training. Yin et al. [122] offer useful insight in this regard. They use a GAN's discriminator to classify the behavior of botnets in netflow datasets. The authors track the precision, accuracy, and F1-Score of the GAN as a function of the number of generated samples used to train it, and compare this against the baseline with no generated samples. In each instance, the graphs showed that while increasing the amount of synthetic data from zero would increase performance, this could only work for so many samples before performance actually started to decline. At around 8,000 samples (compared to around 491,000 in the original dataset), performance was worse than the baseline.

Although it is possible that this particular GAN could benefit from the stabilizing techniques employed in works such as those of Kim et al. [47, 48] and Kargaard et al. [42] to perhaps increase the number of samples before degradation occurs, it nevertheless suggests that GANs may have limits on their utility toward producing security data.

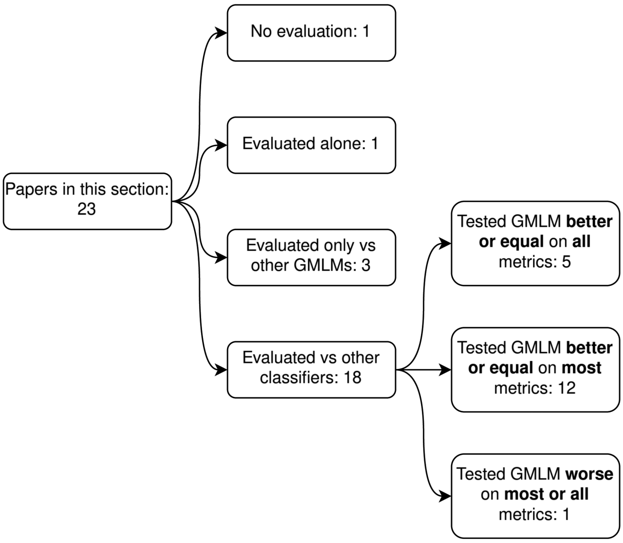

Papers in this section: 23
No evaluation: 1
Evaluated alone: 1
Evaluated only vs other GMLMs: 3
Evaluated vs other classifiers: 18
Tested GMLM **better or equal** on **all** metrics: 5
Tested GMLM **better or equal** on **most** metrics: 12
Tested GMLM **worse** on **most or all** metrics: 1

Fig. 12. Overview of how the papers reviewed under the GMLM as IDS application area are evaluated relative to other models.

8.5 Discussion

Although the primary purpose of a GMLM is for generating data, the algorithms used for training them often lend to the creation of classifier models that can be used for tasks such as intrusion detection. These models, despite being a by-product of training a model whose purpose is not classification, have shown results that are often competitive with traditional IDSs.

Figure 12 presents a taxonomy of how each of the papers presented in this section are evaluated relative to other models. While a small number are either not evaluated, evaluated without other models to compare to, or only compared to other GMLMs, the majority are compared to one or more solutions which do not involve GMLMs. Of those that remain, roughly one-third show results that are, on all metrics, better than every model they have been compared to, or in the worst case, tie on one or more metrics. For the majority of works, it has been found that the GMLM performs best on the majority of metrics but may be beaten by a traditional classifier on at least one metric. In only one case [75] was it found that the GMLM had not performed better than its baseline.

9 CONCLUSION

9.1 Summary

Recent developments in GMLMs have made significant impacts in a number of fields. Among these fields, and of considerable interest, are IDSs. Prior work exploring the application of machine learning to intrusion detection has found that issues with the availability and quality of data, as well as issues with performance, pose an obstacle to their development and future adoption. GMLMs are increasingly providing solutions to these problems. Improving data access is perhaps an intuitively appealing application of GMLMs since, at a high level, the task is straightforward: create better data. However, when more low level details are considered, the task becomes much more complex than it initially appears. In terms of performance issues, these are addressed either by using the GMLM in a supporting role with an existing IDS, such as by creating adversarial data to uncover weaknesses (and improve upon them), or by using the non-generative elements of GMLM architectures to implement new forms of IDSs. In this article, we provided a systematic

Table 4. Open Problems for GMLMs Applied to Intrusion Detection

Pr. No.	Problem Description	Related Sections
1.	Establish standard metrics for tasks where GMLMs are used to improve IDS datasets.	5.3, 7.4
2.	Conduct more research on the use of GMLMs for exploratory attacks.	6.4
3.	Demonstrate the successful execution of a cyber attack, modified by a GMLM to evade an IDS, which accomplishes the same tasks as the unmodified attack.	6.5
4.	Explore further how IDS performance can be improved when vulnerabilities are revealed through GMLM-based penetration testing.	6.5
5.	Explore the effectiveness of GMLMs other than GANs for penetration testing tasks.	6.5
6.	Explore the effectiveness of autoregressive and diffusion models in greater depth for IDS-related tasks.	–
7.	Develop methods for generating sequential alert data.	7.3
8.	Apply GMLMs to the creation of publicly available IDS datasets where GMLM performance is already strong (e.g., FDI attacks).	7.5
9.	Establish standard feature representations for intrusion detection data.	7.5
10.	Find minimums on the amount of data points for minority classes when improving unbalanced datasets.	7.5

mapping study and an in-depth analysis of works to show how IDS-related issues are being addressed, in part, by GMLMs. Guiding our mapping study, we posed two research questions: (1) How are GMLMs used to improve IDS testing? and (2) How are GMLMs used to improve IDS training?

In answering these questions, we uncovered three application areas for GMLMs in the problem domain of intrusion detection: penetration testing, supplementing IDS datasets, and using GMLMs as IDSs. In each of these application areas, we found some degree of success for GMLMs. For penetration testing, we found that GMLMs are effective at finding adversarial samples which will not be detected correctly by IDSs. Compared to the other application areas, however, this topic is under-explored and should receive greater attention. For supplementing IDS datasets, we found that GMLMs are effective at producing new samples of minority classes within datasets, and that IDSs that are trained on these new datasets show performance improvements over those trained on the unbalanced datasets. We also uncovered some issues related to standardization for feature extraction and evaluation in this application area. Finally, where GMLMs are used as IDSs, we found that they often perform as well as, or better than, traditional classifier models at detecting cyber attacks.

9.2 Open Problems

Thus far, existing work with GMLMs in intrusion detection has established their capabilities for improving performance and alleviating issues related to dataset quality and availability. From our analysis of these works, we found that there is potential for greater improvement. This will require further research on a number of open problems. We conclude this article by listing in Table 4 the 10 open problems that we believe are the most significant. Many of these problems were introduced and their contexts described in the “Discussion” subsections in each of Sections 6 through 8. We include references to those sections in the list in Table 4 wherever applicable.

These open problems focus primarily on the first two application areas of penetration testing and supplementing IDS datasets. Although there is room for improvement for GMLMs used as IDSs, their effectiveness in this domain has been demonstrated relatively adequately. Further research on advancing GMLMs as IDSs should consider working toward Problem 6, which is applicable to all three application areas.

The descriptions of most of the problems listed in Table 4 are self-explanatory. Problems 1 and 9 warrant further elaboration, as they concern standards. For Problem 1, we must consider that any metric or combination of metrics should not only show that the synthetic data can improve IDS performance (as TSTR-based metrics necessarily do) but also show that the data is realistic and could appear in real-world cyber monitoring data. For Problem 9, standards will need to be tailored to specific types of intrusion detection data, as what works for netflows may not work for a dataset of malware samples.

REFERENCES

[1] Adel Abusitta, Miles Q. Li, and Benjamin C. M. Fung. 2021. Malware classification and composition analysis: A survey of recent developments. *Journal of Information Security and Applications* 59 (2021), 102828. https://doi.org/10.1016/j.jisa.2021.102828

[2] Silvia T. AcuÃśa, John W. Castro, Oscar Dieste, and Natalia Juristo. 2012. A systematic mapping study on the open source software development process. In *Proceedings of the16th International Conference on Evaluation and Assessment in Software Engineering (EASE'12)*. 42–46. https://doi.org/10.1049/ic.2012.0005

[3] S. Ahmadian, H. Malki, and Z. Han. 2018. Cyber attacks on smart energy grids using generative adversarial networks. In *Proceedings of the 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP'18)*. 942–946.

[4] Mohiuddin Ahmed and Al-Sakib Khan Pathan. 2020. False Data Injection Attack (FDIA): An overview and new metrics for fair evaluation of its countermeasure. *Complex Adaptive Systems Modeling* 8, 1 (April 2020), 4. https://doi.org/10.1186/s40294-020-00070-w

[5] Samet AkÃğay, Amir Atapour-Abarghouei, and Toby P. Breckon. 2019. Skip-GANomaly: Skip connected and adversarially trained encoder-decoder anomaly detection. In *Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN'19)*. 1–8. https://doi.org/10.1109/IJCNN.2019.8851808

[6] Jinwon An and Sungzoon Cho. 2015. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE* 2, 1 (2015), 1–18.

[7] James P. Anderson. 1980. *Computer Security Threat Monitoring and Surveillance*. Technical Report. James P. Anderson Company.

[8] Andy Applebaum, Doug Miller, Blake Strom, Chris Korban, and Ross Wolf. 2016. Intelligent, automated red team emulation. In *Proceedings of the 32nd Annual Conference on Computer Security Applications (ACSAC'16)*. ACM, New York, NY, USA, 363–373. https://doi.org/10.1145/2991079.2991111

[9] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*. 214–223. http://proceedings.mlr.press/v70/arjovsky17a.html

[10] Rafael Ramos Regis Barbosa and Aiko Pras. 2010. Intrusion detection in SCADA networks. In *Mechanisms for Autonomous Management of Networks and Services*, Burkhard Stiller and Filip De Turck (Eds.). Springer, Berlin, Germany, 163–166.

[11] David Berthelot, Tom Schumm, and Luke Metz. 2017. BEGAN: Boundary equilibrium generative adversarial networks. *arXiv abs/1703.10717* (2017).

[12] Alceu Bissoto, Eduardo Valle, and Sandra Avila. 2021. GAN-based data augmentation and anonymization for skin-lesion analysis: A critical review. In *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'21)*. 1847–1856. https://doi.org/10.1109/CVPRW53098.2021.00204

[13] Ali Borji. 2022. Generated faces in the wild: Quantitative comparison of stable diffusion, midjourney and DALL-E 2. *arXiv:2210.00586* (2022). https://doi.org/10.48550/ARXIV.2210.00586

[14] Daniela Brauckhoff, Arno Wagner, and Martin May. 2008. FLAME: A flow-level anomaly modeling engine. In *Proceedings of the Conference on Cyber Security Experimentation and Test*.

[15] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford,

Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, 1877–1901. https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[16] Keyang Cheng, Rabia Tahir, Lubamba Kasangu Eric, and Maozhen Li. 2020. An analysis of generative adversarial networks and variants for image synthesis on MNIST dataset. *Multimedia Tools and Applications* 79, 19 (May 2020), 13725–13752. https://doi.org/10.1007/s11042-019-08600-2

[17] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. 2022. Diffusion models in vision: A survey. *arXiv:2209.04747* (2022). https://doi.org/10.48550/ARXIV.2209.04747

[18] Tuan Van Dao, Hiroshi Sato, and Masao Kubo. 2022. An attention mechanism for combination of CNN and VAE for image-based malware classification. *IEEE Access* 10 (2022), 85127–85136. https://doi.org/10.1109/ACCESS.2022.3198072

[19] Hans de Bruijn and Marijn Janssen. 2017. Building cybersecurity awareness: The need for evidence-based framing strategies. *Government Information Quarterly* 34, 1 (2017), 1–7. https://doi.org/10.1016/j.giq.2017.02.007

[20] Alfonso Delgado-Bonal and Alexander Marshak. 2019. Approximate entropy and sample entropy: A comprehensive tutorial. *Entropy* 21, 6 (2019), 541. https://doi.org/10.3390/e21060541

[21] Phai Vu Dinh, Nguyen Quang Uy, Diep N. Nguyen, Dinh Thai Hoang, Son Pham Bao, and Eryk Dutkiewicz. 2022. Twin variational auto-encoder for representation learning in IoT intrusion detection. In *Proceedings of the 2022 IEEE Wireless Communications and Networking Conference (WCNC'22)*. 848–853. https://doi.org/10.1109/WCNC51071.2022.9771793

[22] Carl Doersch. 2016. Tutorial on variational autoencoders. *arXiv:1606.05908* (2016). https://doi.org/10.48550/ARXIV.1606.05908

[23] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. 2017. Adversarial feature learning. In *Proceedings of the 5th International Conference on Learning Representations (ILCR'17): Conference Track*. https://openreview.net/forum?id=BJtNZAFgg

[24] Indira Kalyan Dutta, Bhaskar Ghosh, Albert Carlson, Michael Totaro, and Magdy Bayoumi. 2020. Generative adversarial networks in security: A survey. In *Proceedings of the 2020 11th IEEE Annual Ubiquitous Computing, Electronics, and Mobile Communication Conference (UEMCON'20)*. 0399–0405. https://doi.org/10.1109/UEMCON51285.2020.9298135

[25] Aidin Ferdowsi and Walid Saad. 2019. Generative adversarial networks for distributed intrusion detection in the Internet of Things. In *Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM'19)*. 1–6. https://doi.org/10.1109/GLOBECOM38437.2019.9014102

[26] Ugo Fiore, Alfredo De Santis, Francesca Perla, Paolo Zanetti, and Francesco Palmieri. 2019. Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. *Information Sciences* 479 (2019), 448–455. https://doi.org/10.1016/j.ins.2017.12.030

[27] Paulo Freitas de Araujo-Filho, Georges Kaddoum, Divanilson R. Campelo, Aline Gondim Santos, David MacÃłdo, and Cleber Zanchettin. 2021. Intrusion detection for cyber-physical systems using generative adversarial networks in fog environment. *IEEE Internet of Things Journal* 8, 8 (2021), 6247–6256. https://doi.org/10.1109/JIOT.2020.3024800

[28] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems—Volume 2 (NIPS'14)*. 2672–2680.

[29] Karol Gregor, Ivo Danihelka, Andriy Mnih, Charles Blundell, and Daan Wierstra. 2014. Deep autoregressive networks. In *Proceedings of the 31st International Conference on Machine Learning*. 1242–1250. https://proceedings.mlr.press/v32/gregor14.html

[30] Michael Guastalla, Yiyi Li, Arvin Hekmati, and Bhaskar Krishnamachari. 2024. Application of large language models to DDoS attack detection. In *Security and Privacy in Cyber-Physical Systems and Smart Vehicles*, Yu Chen, Chung-Wei Lin, Bo Chen, and Qi Zhu (Eds.). Springer Nature Switzerland, Cham, 83–99.

[31] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. 2017. Improved training of Wasserstein GANs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*. 5769–5779.

[32] Nutan Farah Haq, Abdur Rahman Onik, Md. Avishek Khan Hridoy, Musharrat Rafni, Faisal Muhammad Shah, and Dewan Md. Farid. 2015. Application of machine learning approaches in intrusion detection system: A survey. *International Journal of Advanced Research in Artificial Intelligence* 4, 3 (2015), 9–18. https://doi.org/10.14569/IJARAI.2015.040302

[33] L. T. Heberlein, G. V. Dias, K. N. Levitt, B. Mukherjee, J. Wood, and D. Wolber. 1990. A network security monitor. In *Proceedings of the 1990 IEEE Computer Society Symposium on Research in Security and Privacy*. 296–304. https://doi.org/10.1109/RISP.1990.63859

[34] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, 6840–6851. https://proceedings.neurips.cc/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf

[35] Joerg Hoffmann. 2015. Simulated Penetration Testing: From "Dijkstra" to "Turing Test++." RetrievedApril25, 2024fromhttps://www.aaai.org/ocs/index.php/ICAPS/ICAPS15/paper/view/10495

[36] Weiwei Hu and Ying Tan. 2017. Generating adversarial malware examples for black-box attacks based on GAN. *arXiv:1702.05983 [cs.LG]* (2017).

[37] Håkon Hukkelås, Rudolf Mester, and Frank Lindseth. 2019. DeepPrivacy: A generative adversarial network for face anonymization. In *Advances in Visual Computing*. Springer International Publishing, Cham, 565–578.

[38] Idriss Idrissi, Mostafa Azizi, and Omar Moussaoui. 2022. An unsupervised generative adversarial network based-host intrusion detection system for Internet of Things devices. *Indonesian Journal of Electrical Engineering and Computer Science* 25, 2 (2022), 1140–1150.

[39] Meryem Janati Idrissi, Hamza Alami, Abdelhak Bouayad, and Ismail Berrada. 2023. NF-NIDS: Normalizing Flows for Network Intrusion Detection Systems. In *Proceedings of the 2023 10th International Conference on Wireless Networks and Mobile Communications (WINCOM'23)*. 1–7. https://doi.org/10.1109/WINCOM59760.2023.10322987

[40] Salman Jan, Shahrulniza Musa, Toqeer Syed, and Ali Alzahrani. 2018. Deep convolutional generative adversarial networks for in-tent-based dynamic behavior capture. *International Journal of Engineering and Technology* 7 (2018), 101–103.

[41] Runhai Jiao, Gangyi Xun, Xuan Liu, and Guangwei Yan. 2021. A new AC false data injection attack method without network information. *IEEE Transactions on Smart Grid* 12, 6 (2021), 5280–5289. https://doi.org/10.1109/TSG.2021.3102329

[42] J. Kargaard, T. Drange, A. Kor, H. Twafik, and E. Butterfield. 2018. Defending IT systems against intelligent malware. In *Proceedings of the 2018 IEEE 9th International Conference on Dependable Systems, Services, and Technologies (DESSERT'18)*. 411–417.

[43] M. Kawai, K. Ota, and M. Dong. 2019. Improved MalGAN: Avoiding malware detector by leaning cleanware features. In *Proceedings of the 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC'19)*. 040–045.

[44] Shapla Khanam, Ismail Ahmedy, Mohd Yamani Idna Idris, and Mohamed Hisham Jaward. 2022. Towards an effective intrusion detection model using focal loss variational autoencoder for internet of things (IoT). *Sensors* 22, 15 (2022), 5822. https://doi.org/10.3390/s22155822

[45] Ansam Khraisat, Iqbal Gondal, Peter Vamplew, and Joarder Kamruzzaman. 2019. Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity* 2, 1 (July 2019), 20. https://doi.org/10.1186/s42400-019-0038-7

[46] Hyeongju Kim, Hyeonseung Lee, Woo Hyun Kang, Sung Jun Cheon, Byoung Jin Choi, and Nam Soo Kim. 2020. WaveNODE: A continuous normalizing flow for speech synthesis. *arXiv:2006.04598 [cs.SD]* (2020).

[47] Jin-Young Kim, Seok-Jun Bu, and Sung-Bae Cho. 2017. Malware detection using deep transferred generative adversarial networks. In *Neural Information Processing*, Derong Liu, Shengli Xie, Yuanqing Li, Dongbin Zhao, and El-Sayed M. El-Alfy (Eds.). Springer International Publishing, Cham, 556–564.

[48] Jin-Young Kim, Seok-Jun Bu, and Sung-Bae Cho. 2018. Zero-day malware detection using transferred generative adversarial networks based on deep autoencoders. *Information Sciences* 460-461 (2018), 83–102. https://doi.org/10.1016/j.ins.2018.04.092

[49] Diederik P. Kingma and Max Welling. 2013. Auto-encoding variational Bayes. *arXiv:1312.6114* (2013). https://doi.org/10.48550/ARXIV.1312.6114

[50] Sharmila KishorWagh, Vinod Pachghare, and Satish Kolhe. 2013. Survey on intrusion detection system using machine learning techniques. *International Journal of Computer Applications* 78 (2013), 30–37. https://doi.org/10.5120/13608-1412

[51] Ivan Kobyzev, Simon Prince, and Marcus A. Brubaker. 2019. Normalizing Flows: Introduction and ideas. *Stat* 1050 (2019), 25.

[52] Manoj Kumar, Mohammad Babaeizadeh, Dumitru Erhan, Chelsea Finn, Sergey Levine, Laurent Dinh, and Durk Kingma. 2019. Videoflow: A flow-based generative model for video. *arXiv preprint arXiv:1903.01434* (2019).

[53] David Kushner. 2013. The real story of Stuxnet: How Kaspersky Lab tracked down the malware that stymied Iran's nuclear-fuel enrichment program. *IEEE Spectrum*. Retrieved November 24, 2019 from https://spectrum.ieee.org/telecom/security/the-real-story-of-stuxnet

[54] Yifan Li, Xiaoyan Peng, Ziyan Wu, Fan Yang, Xuan He, and Zhiyong Li. 2023. M3GAN: A masking strategy with a mutable filter for multidimensional anomaly detection. *Knowledge-Based Systems* 271 (2023), 110585. https://doi.org/10.1016/j.knosys.2023.110585

[55] G. Liang, S. R. Weller, J. Zhao, F. Luo, and Z. Y. Dong. 2017. The 2015 Ukraine blackout: Implications for false data injection attacks. *IEEE Transactions on Power Systems* 32, 4 (2017), 3317–3318.

[56] Hung-Jen Liao, Chun-Hung Richard Lin, Ying-Chih Lin, and Kuang-Yuan Tung. 2013. Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications* 36, 1 (2013), 16–24. https://doi.org/10.1016/j.jnca.2012.09.004

[57] Ying-Dar Lin, Zi-Qiang Liu, Ren-Hung Hwang, Van-Linh Nguyen, Po-Ching Lin, and Yuan-Cheng Lai. 2022. Machine learning with variational autoencoder for imbalanced datasets in intrusion detection. *IEEE Access* 10 (2022), 15247–15260. https://doi.org/10.1109/ACCESS.2022.3149295

[58] Zilong Lin, Yong Shi, and Zhi Xue. 2022. IDSGAN: Generative adversarial networks for attack generation against intrusion detection. In *Advances in Knowledge Discovery and Data Mining*, João Gama, Tianrui Li, Yang Yu, Enhong Chen, Yu Zheng, and Fei Teng (Eds.). Springer International Publishing, Cham, 79–91.

[59] Hongyu Liu and Bo Lang. 2019. Machine learning and deep learning methods for intrusion detection systems: A survey. *Applied Sciences* 9, 20 (2019), 4396. https://doi.org/10.3390/app9204396

[60] Manuel Lopez-Martin, Belen Carro, and Antonio Sanchez-Esguevillas. 2019. Variational data generative model for intrusion detection. *Knowledge and Information Systems* 60, 1 (July 2019), 569–590. https://doi.org/10.1007/s10115-018-1306-7

[61] Manuel Lopez-Martin, Belen Carro, Antonio Sanchez-Esguevillas, and Jaime Lloret. 2017. Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in IoT. *Sensors* 17, 9 (2017), 1967. https://doi.org/10.3390/s17091967

[62] Teresa F. Lunt, R. Jagannathan, Rosanna Lee, Sherry Listgarten, David L. Edwards, Peter G. Neumann, Herald S. Javitz, and L. Valdes. 1988. *IDES: The Enhanced Prototype. A Real-Time Intrusion Detection Expert System*. SRI International. Computer Science Laboratory.

[63] Liam Daly Manocchio, Siamak Layeghy, and Marius Portmann. 2021. FlowGAN—Synthetic network flow generation using generative adversarial networks. In *Proceedings of the 2021 IEEE 24th International Conference on Computational Science and Engineering (CSE'21)*. 168–176. https://doi.org/10.1109/CSE53436.2021.00033

[64] Gerard MartÃŋ-Juan, Marco Lorenzi, and Gemma Piella. 2023. MC-RVAE: Multi-channel recurrent variational autoencoder for multimodal Alzheimer's disease progression modelling. *NeuroImage* 268 (2023), 119892. https://doi.org/10.1016/j.neuroimage.2023.119892

[65] Marek MaÅĆowidzki, Przemyslaw Berezinski, and MichaÅĆ Mazur. 2015. Network intrusion detection: Half a kingdom for a good dataset. In *Proceedings of the NATO STO SAS-139 Workshop*.

[66] M. L. MenÃĺndez, J. A. Pardo, L. Pardo, and M. C. Pardo. 1997. The Jensen-Shannon divergence. *Journal of the Franklin Institute* 334, 2 (1997), 307–318. https://doi.org/10.1016/S0016-0032(96)00063-4

[67] Tim Merino, Matt Stillwell, Mark Steele, Max Coplan, Jon Patton, Alexander Stoyanov, and Lin Deng. 2020. *Expansion of Cyber Attack Data from Unbalanced Datasets Using Generative Adversarial Networks*. Springer International Publishing, Cham, 131–145. https://doi.org/10.1007/978-3-030-24344-9_8

[68] Mostafa Mohammadpourfard, Fateme Ghanaatpishe, Marziyeh Mohammadi, Subhash Lakshminarayana, and Mykola Pechenizkiy. 2020. Generation of false data injection attacks using conditional generative adversarial networks. In *Proceedings of the 2020 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe'20)*. 41–45. https://doi.org/10.1109/ISGT-Europe47291.2020.9248967

[69] Nour Moustafa and Jill Slay. 2015. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS'15)*. 1–6. https://doi.org/10.1109/MilCIS.2015.7348942

[70] Naji Najari, Samuel Berlemont, GrÃĺgoire Lefebvre, Stefan Duffner, and Christophe Garcia. 2022. Robust variational autoencoders and normalizing flows for unsupervised network anomaly detection. In *Advanced Information Networking and Applications*, Leonard Barolli, Farookh Hussain, and Tomoya Enokido (Eds.). Springer International Publishing, Cham, 281–292.

[71] Mohammad Nauman, Tamleek Tanveer, Sohail Khan, and Toqeer Syed. 2018. Deep neural architectures for large scale Android malware analysis. *Cluster Computing* 21 (2018), 569–588. https://doi.org/10.1007/s10586-017-0944-y

[72] Laisen Nie, Yixuan Wu, Xiaojie Wang, Lei Guo, Guoyin Wang, Xinbo Gao, and Shengtao Li. 2022. Intrusion detection for secure Social Internet of Things based on collaborative edge computing: A generative adversarial network-based approach. *IEEE Transactions on Computational Social Systems* 9, 1 (2022), 134–145. https://doi.org/10.1109/TCSS.2021.3063538

[73] Skyler Norgaard, Ramyar Saeedi, Keyvan Sasani, and A. H. Gebremedhin. 2018. Synthetic sensor data generation for health applications: A supervised deep learning approach. In *Proceedings of the 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC'18)*.

[74] OpenAI. 2023. GPT-4 technical report. *arXiv:2303.08774 [cs.CL]* (2023).

[75] Genki Osada, Kazumasa Omote, and Takashi Nishide. 2017. Network intrusion detection based on semi-supervised variational auto-encoder. In *Computer Security—ESORICS 2017*, Simon N. Foley, Dieter Gollmann, and Einar Snekkenes (Eds.). Springer International Publishing, Cham, 344–361.

[76] K. Pan, A. Teixeira, M. Cvetkovic, and P. Palensky. 2019. Cyber risk analysis of combined data attacks against power system state estimation. *IEEE Transactions on Smart Grid* 10, 3 (2019), 3044–3056.

[77] Nilesh Pandey and Andreas E. Savakis. 2019. Poly-GAN: Multi-conditioned GAN for fashion synthesis. *Neurocomputing* 414 (2019), 356–364. https://api.semanticscholar.org/CorpusID:202539671

[78] Birgit Penzenstadler, Ankita Raturi, Debra Richardson, Coral Calero, Henning Femmer, and Xavier Franch. 2014. Systematic mapping study on software engineering for sustainability (SE4S). In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering (EASE'14)*. ACM, New York, NY, USA, Article 14, 14 pages. https://doi.org/10.1145/2601248.2601256

[79] Kai Petersen, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. 2008. Systematic mapping studies in software engineering. In *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering (EASE'08)*. 68–77.

[80] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with CLIP latents. *arXiv:2204.06125* (2022). https://doi.org/10.48550/ARXIV.2204.06125

[81] M. Rigaki and S. Garcia. 2018. Bringing a GAN to a knife-fight: Adapting malware communication to avoid detection. In *Proceedings of the 2018 IEEE Security and Privacy Workshops (SPW'18)*. 70–75.

[82] Markus Ring, Daniel SchlÃűr, Dieter Landes, and Andreas Hotho. 2019. Flow-based network traffic generation using generative adversarial networks. *Computers & Security* 82 (2019), 156–172. https://doi.org/10.1016/j.cose.2018.12.012

[83] Markus Ring, Sarah Wunderlich, Dominik GrÃijdl, Dieter Landes, and Andreas Hotho. 2017. Flow-based benchmark data sets for intrusion detection. In *Proceedings of the European Conference on Cyber Warfare and Security (ECCWS'17)*. 1–10.

[84] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10684–10695.

[85] Royi Ronen, Marian Radu, Corina Feuerstein, Elad Yom-Tov, and Mansour Ahmadi. 2018. Microsoft Malware Classification Challenge. *arXiv:1802.10135* (2018). https://doi.org/10.48550/ARXIV.1802.10135

[86] Sanjay K. Sahay, Ashu Sharma, and Hemant Rathore. 2020. Evolution of malware and its detection techniques. In *Information and Communication Technology for Sustainable Development*, Milan Tuba, Shyam Akashe, and Amit Joshi (Eds.). Springer Singapore, Singapore, 139–150.

[87] Milad Salem, Shayan Taheri, and Jiann Shiun Yuan. 2018. Anomaly generation using generative adversarial networks in host-based intrusion detection. In *Proceedings of the 2018 9th IEEE Annual Ubiquitous Computing, Electronics, and Mobile Communication Conference (UEMCON'18)*. IEEE. https://doi.org/10.1109/uemcon.2018.8796769

[88] Mustafizur R. Shahid, Gregory Blanc, Houda Jmila, Zonghua Zhang, and HervÃľ Debar. 2020. Generative deep learning for Internet of Things network traffic generation. In *Proceedings of the 2020 IEEE 25th Pacific Rim International Symposium on Dependable Computing (PRDC'20)*. 70–79. https://doi.org/10.1109/PRDC50213.2020.00018

[89] M. Shahriar, N. Haque, M. Rahman, and M. Alonso. 2020. G-IDS: Generative adversarial networks assisted intrusion detection system. In *Proceedings of the 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC'20)*. IEEE, 376–385. https://doi.org/10.1109/COMPSAC48688.2020.0-218

[90] Md. Hasan Shahriar, Alvi Ataur Khalil, Mohammad Ashiqur Rahman, Mohammad Hossein Manshaei, and Dong Chen. 2021. iAttackGen: Generative synthesis of false data injection attacks in cyber-physical systems. In *Proceedings of the 2021 IEEE Conference on Communications and Network Security (CNS'21)*. 200–208. https://doi.org/10.1109/CNS53000.2021.9705034

[91] Wenling Shang, Kihyuk Sohn, and Yuandong Tian. 2018. Channel-recurrent autoencoding for image modeling. In *Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV'18)*. 1195–1204. https://doi.org/10.1109/WACV.2018.00136

[92] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. 2018. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *Proceedings of the 4th International Conference on Information Systems Security and Privacy.*

[93] Yi Shi, E. Yalin Sagduyu, Kemal Davaslioglu, and H. Jason Li. 2018. Generative adversarial networks for black-box API attacks with limited training data. In *Proceedings of the 2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT'18)*. 453–458.

[94] Geeta Singh and Neelu Khare. 2022. A survey of intrusion detection from the perspective of intrusion datasets and machine learning techniques. *International Journal of Computers and Applications* 44, 7 (2022), 659–669. https://doi.org/10.1080/1206212X.2021.1885150

[95] Zhanna Malekos Smith, Eugenia Lostri, and James A Lewis. 2020. The hidden costs of cybercrime. *Trellix*. Retrieved April 25, 2024 from https://www.mcafee.com/enterprise/en-us/assets/reports/rp-hidden-costs-of-cybercrime.pdf

[96] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Eds.). Vol. 28. Curran Associates, 1–9. https://proceedings.neurips.cc/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf

[97] Jacob Steinhardt, Pang Wei Koh, and Percy Liang. 2017. Certified defenses for data poisoning attacks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*. 3520–3532.

[98] Sungho Suh, Haebom Lee, Paul Lukowicz, and Yong Oh Lee. 2021. CEGAN: Classification enhancement generative adversarial networks for unraveling data imbalance problems. *Neural Networks* 133 (2021), 69–86. https://doi.org/10.1016/j.neunet.2020.10.004

[99] Jiayu Sun, Xinzhou Wang, Naixue Xiong, and Jie Shao. 2018. Learning sparse representation with variational autoencoder for anomaly detection. *IEEE Access* 6 (2018), 33353–33361. https://doi.org/10.1109/ACCESS.2018.2848210

[100] X. Sun, J. Dai, P. Liu, A. Singhal, and J. Yen. 2018. Using Bayesian networks for probabilistic identification of zero-day attack paths. *IEEE Transactions on Information Forensics and Security* 13, 10 (2018), 2506–2521.

[101] Michael J. Swain and Dana H. Ballard. 1991. Color indexing. *International Journal of Computer Vision* 7, 1 (Nov. 1991), 11–32. https://doi.org/10.1007/BF00130487

[102] Christopher Sweet. 2019. *Synthesizing Cyber Intrusion Alerts Using Generative Adversarial Networks*. Master's Thesis. Rochester Institute of Technology.

[103] C. Sweet, S. Moskal, and S. J. Yang. 2019. Synthetic intrusion alert generation through generative adversarial networks. In *Proceedings of the 2019 IEEE Military Communications Conference (MILCOM'19)*. 1–6.

[104] Christopher Sweet, Stephen Moskal, and Shanchieh Jay Yang. 2020. On the variety and veracity of cyber intrusion alerts synthesized by generative adversarial networks. *ACM Transactions on Management Information Systems* 11, 4 (Oct. 2020), Article 22, 21 pages. https://doi.org/10.1145/3394503

[105] Wesley Tann, Yuancheng Liu, Jun Heng Sim, Choon Meng Seah, and Ee-Chien Chang. 2023. Using large language models for cybersecurity Capture-The-Flag challenges and certification questions. *arXiv:2308.10443 [cs.AI]* (2023).

[106] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani. 2009. A detailed analysis of the KDD CUP 99 data set. In *Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*. 1–6. https://doi.org/10.1109/CISDA.2009.5356528

[107] Thomas Taylor and Amna Eleyan. 2021. Using variational autoencoders to increase the performance of malware classification. In *Proceedings of the 2021 International Symposium on Networks, Computers, and Communications (ISNCC'21)*. 1–6. https://doi.org/10.1109/ISNCC52172.2021.9615643

[108] Xue Tong and Wang Qi. 2021. False data injection attack on power system data-driven methods based on generative adversarial networks. In *Proceedings of the 2021 IEEE Sustainable Power and Energy Conference (iSPEC'21)*. 4250–4254. https://doi.org/10.1109/iSPEC53008.2021.9735442

[109] M. Usama, M. Asim, S. Latif, J. Qadir, and Ala-Al-Fuqaha. 2019. Generative adversarial networks for launching and thwarting adversarial attacks on network intrusion detection systems. In *Proceedings of the 2019 15th International Wireless Communications Mobile Computing Conference (IWCMC'19)*. 78–83.

[110] Arash Vahdat and Jan Kautz. 2020. NVAE: A deep hierarchical variational autoencoder. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS'20)*. Article 1650, 13 pages.

[111] Aaron van den Oord, Nal Kalchbrenner, Lasse Espeholt, Koray Kavukcuoglu, Oriol Vinyals, and Alex Graves. 2016. Conditional image generation with PixelCNN decoders. In *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (Eds.), Vol. 29. Curran Associates, 1–9. https://proceedings.neurips.cc/paper/2016/file/b1301141feffabac455e1f90a7de2054-Paper.pdf

[112] R. Vinayakumar, Mamoun Alazab, K. P. Soman, Prabaharan Poornachandran, Ameer Al-Nemrat, and Sitalakshmi Venkatraman. 2019. Deep learning approach for intelligent intrusion detection system. *IEEE Access* 7 (2019), 41525–41550. https://doi.org/10.1109/ACCESS.2019.2895334

[113] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F. Wang. 2017. Generative adversarial networks: Introduction and outlook. *IEEE/CAA Journal of Automatica Sinica* 4, 4 (2017), 588–598.

[114] Shengzhe Xu, Manish Marwah, Martin Arlitt, and Naren Ramakrishnan. 2021. STAN: Synthetic network traffic generation with generative neural models. In *Deployable Machine Learning for Security Defense*, Gang Wang, Arridhana Ciptadi, and Ali Ahmadzadeh (Eds.). Springer International Publishing, Cham, 3–29.

[115] Xing Xu, Jie Li, Yang Yang, and Fumin Shen. 2021. Toward effective intrusion detection using log-cosh conditional variational autoencoder. *IEEE Internet of Things Journal* 8, 8 (2021), 6187–6196. https://doi.org/10.1109/JIOT.2020.3034621

[116] Tijin Yan, Tong Zhou, Yufeng Zhan, and Yuanqing Xia. 2021. TFDPM: Attack detection for cyber-physical systems with diffusion probabilistic models. *arXiv:2112.10774* (2021). https://doi.org/10.48550/ARXIV.2112.10774

[117] Jian Yang, Xiang Chen, Shuangwu Chen, Xiaofeng Jiang, and Xiaobin Tan. 2021. Conditional variational auto-encoder and extreme value theory aided two-stage learning approach for intelligent fine-grained known/unknown intrusion detection. *IEEE Transactions on Information Forensics and Security* 16 (2021), 3538–3553. https://doi.org/10.1109/TIFS.2021.3083422

[118] Li-Chia Yang, Szu-Yu Chou, and Yi-Hsuan Yang. 2017. MidiNet: A convolutional generative adversarial network for symbolic-domain music generation using 1D and 2D conditions. *arXiv:1703.10847* (2017).

[119] Yanqing Yang, Kangfeng Zheng, Bin Wu, Yixian Yang, and Xiujuan Wang. 2020. Network intrusion detection based on supervised adversarial variational auto-encoder with regularization. *IEEE Access* 8 (2020), 42169–42184. https://doi.org/10.1109/ACCESS.2020.2977007

[120] Yanqing Yang, Kangfeng Zheng, Chunhua Wu, and Yixian Yang. 2019. Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network. *Sensors* 19, 11 (2019), 2528. https://doi.org/10.3390/s19112528

[121] Ibrahim Yilmaz, Rahat Masum, and Ambareen Siraj. 2020. Addressing imbalanced data problem with generative adversarial network for intrusion detection. In *Proceedings of the 2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI'20)*. 25–30. https://doi.org/10.1109/IRI49571.2020.00012

[122] C. Yin, Y. Zhu, S. Liu, J. Fei, and H. Zhang. 2018. An enhancing framework for botnet detection using generative adversarial networks. In *Proceedings of the 2018 International Conference on Artificial Intelligence and Big Data (ICAIBD'18)*. 228–234.

[123] Yucheng Yin, Zinan Lin, Minhao Jin, Giulia Fanti, and Vyas Sekar. 2022. Practical GAN-based synthetic IP header trace generation using NetShare. In *Proceedings of the ACM SIGCOMM 2022 Conference (SIGCOMM'22)*. ACM, New York, NY, USA, 458–472. https://doi.org/10.1145/3544216.3544251

[124] Chika Yinka-Banjo and Ogban-Asuquo Ugot. 2020. A review of generative adversarial networks and its application in cybersecurity. *Artificial Intelligence Review* 53, 3 (March 2020), 1721–1736. https://doi.org/10.1007/s10462-019-09717-4

[125] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2016. SeqGAN: Sequence generative adversarial nets with policy gradient. *arXiv:1609.05473 [cs.LG]* (2016).

[126] Sultan Zavrak and Murat Äřskefiyeli. 2020. Anomaly-based intrusion detection from network flow features using variational autoencoder. *IEEE Access* 8 (2020), 108346–108358. https://doi.org/10.1109/ACCESS.2020.3001350

[127] H. Zhang, V. Sindagi, and V. M. Patel. 2019. Image de-raining using a conditional generative adversarial network. *IEEE Transactions on Circuits and Systems for Video Technology.* Published Online, June 3, 2019.

[128] Junbo Jake Zhao, Michaël Mathieu, and Yann LeCun. 2017. Energy-based generative adversarial networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR'17): Conference Track*. https://openreview.net/forum?id=ryh9pmcee

[129] Pasquale Zingo and Andrew Novocin. 2020. Can GAN-generated network traffic be used to train traffic anomaly classifiers? In *Proceedings of the 2020 11th IEEE Annual Information Technology, Electronics, and Mobile Communication Conference (IEMCON'20)*. 540–545. https://doi.org/10.1109/IEMCON51383.2020.9284901

Received 16 January 2023; revised 27 February 2024; accepted 31 March 2024