2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA) | 978-1-6654-1623-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/TPSISA52974.2021.00020

Deception for Cyber Defence: Challenges and Opportunities

David Liebowitz[‡§], Surya Nepal[*†], Kristen Moore[*†], Cody J. Christopher[*†], Salil S. Kanhere [§]
David Nguyen[*†§], Roelien C. Timmer[*†§], Michael Longland[*†§], Keerth Rathakumar[*†§]

[*]Data61, CSIRO
Australia
{first.last}@data61.csiro.au

[†]Cyber Security
Cooperative Research
Centre Australia

[‡]Penten Pty Ltd
Canberra, Australia
{first.last}@penten.com

[§]UNSW
Sydney, Australia
{first.last}@unsw.edu.au

***Abstract*—Deception is rapidly growing as an important tool for cyber defence, complementing existing perimeter security measures to rapidly detect breaches and data theft. One of the factors limiting the use of deception has been the cost of generating realistic artefacts by hand. Recent advances in Machine Learning have, however, created opportunities for scalable, automated generation of realistic deceptions. This vision paper describes the opportunities and challenges involved in developing models to mimic many common elements of the IT stack for deception effects.**

***Index Terms*—cyber deception, generative modelling, simulation**

I. Introduction

The digital revolution has touched every aspect of our lives. Precision health, digital agriculture, autonomous vehicles, digital governments and services are a few examples. The COVID-19 pandemic taught us that we could rely on the internet-enabled digital world to do many things that we did not think possible even a year ago. But as the old saying goes, there is no free lunch – and this is especially true in the digital world. There are many challenges, but cybersecurity stands front and centre as one of the most significant. Cybersecurity risks pose a threat to the continued growth and success of the internet-enabled digital world.

Governments, academia and industry have recognised this challenge. An enormous amount of money has been spent on cybersecurity, with forecasts of expenditure exceeding USD 150 billion worldwide in 2021 [1]. Despite these efforts, breaches and data theft continue. Existing cyber defence solutions are like a tailor's patchwork. Most of them are reactive and are not able to deal with sophisticated attacks. The Target Corporation breach of 2013 exemplifies the problem of existing reactive solutions [2].

Target was compromised by a sophisticated attack that resulted in a period of approximately a month with intruders active on their network. This included access to their point-of-sale system over the busy Black Friday shopping period [3]. The breach resulted in the theft of 40 million credit card numbers and 70 million personal records, and an estimated card reissue cost of USD 200 million [4]. Lawsuits have dragged on for years [5] and the event is still used as a cautionary case study [6], [7].

The Target case study gives us two important insights into reactive cyber defence solutions. First, Target's security system raised alarms at the initial breach, but these were reportedly never investigated [8]. The most likely reason for this is simple alarm fatigue. The patchwork-based reactive solutions generate a massive number of alerts. It is impossible to investigate all of them, and the prioritisation of alerts remains a challenge. Second, the damage was compounded by the dwell time. Prolonged, undetected access permitted movement across the network and allowed the sustained harvesting of information. According to the 2021 Cost of a Data Breach Report [9], the *average* time taken to identify a breach has risen to 212 days. We have witnessed several real-world case studies in the last decade with similar attributes and results [2]. The time is upon us to rethink our approaches to cybersecurity and incorporate more proactive cyber defence tools to our existing security arsenal.

In this vision article, we argue that it is time to revisit cyber deception. We believe that this is one of the most potent defensive security tools at our disposal and is arguably one of the most underutilised. Deception is an attempt to manipulate the beliefs of others in order to influence their behaviour. In the cyber domain, this usually means creating and deploying a *honeypot* of some sort - a fake digital resource that mimics some characteristics of real resources [10]. In doing so, the cyber defender tricks intruders, data thieves or malicious insiders into behaviour that reveals their presence and possibly more information.

Take, for example, a honeypot appliance on a network, mimicking a server [11]. It is possible that a systems administrator *might* accidentally try to access it during routine maintenance. However, an access attempt is far more likely due to intruder lateral movement or attempted data theft. This ability to highlight unauthorised activity makes deception a valuable breach discovery mechanism. Its intrinsically low false positive alarm rates can mitigate some of the effects of the alert fatigue problem, which helps solve the breach discovery problem, which in turn solves the problem of dwell time.

In addition to its role in counteracting alert fatigue and reducing dwell time, deceptions can become a mechanism to drive adversary interactions that can provide intelligence

978-1-6654-1623-8/21/$31.00 ©2021 IEEE
DOI 10.1109/TPSISA52974.2021.00020

on their intent, or their tactics, tools and procedures (TTPs). While simpler honeypots can be very effective for breach and theft detection, they do not drive the level of interaction necessary for these additional effects. Simple honeypots can also become familiar to adversaries, who can begin to detect them without raising alarms.

The key to engaging intruders to interact with deceptions is *realism*. Deceptive artefacts must be indistinguishable from the real subjects of mimicry, at least up to a level appropriate to the context and interaction. Generating such realistic honeypots by hand requires a significant amount of tedious work, which has impeded development of deceptive technologies until now. Indeed, we believe that the main challenge is to create deceptions automatically, so that they can be generated and deployed at the scale necessary to be useful as a defensive tool. Recent advances in Machine Learning means it is now possible to model and create artifacts to simulate characteristics of many IT assets and processes. This vision paper presents our work-in-progress on such modeling.

We discuss some of the background to cyber deception in Section II. Section III summarises some challenges and opportunities in using ML. It also highlights and provides a brief overview of our ongoing research and outcomes in this area. Finally, we present some discussions and concluding remarks.

II. Background

What is Deception? Deception is an attempt to manipulate the beliefs of others to influence their behaviour. The practice of deception is as old as life itself. The fierce competition to survive has produced breathtaking examples in the natural world [12], such as camouflage predators like cuttlefish that can change the colour and texture of their skin to match their surroundings, insects indistinguishable from twigs when stationary, eye spots on moths that imitate much larger animals and plants that mimic pheromones or food scents to attract pollinating insects.

Deception in the Physical World: Deception has also long been a key strategy in human conflict and warfare. Operation Bodyguard [13], to take one famous example, was a World War II Allied deception plan implemented to mislead the German high command about the site of the D-Day landings in Normandy. The plan was named for Churchill's observation that "In wartime, truth is so precious that she should always be attended by a bodyguard of lies." Large scale, coordinated activity with dummy vehicles and aircraft, fake radio signals and the use of double agents created the illusion of Allied invasion forces poised to attack various parts of the European coastline. It successfully concealed the real landing site and prompted diversion of defensive forces away from it, making a substantial contribution to Allied victory in the Battle of Normandy and the subsequent end of the war. We can protect information resources with the very same concepts and techniques found in nature and expressed in conflict [14], [15], but adapted to the cyber domain.

Deception in the Cyber Domain: The power of deception as a cyber defensive tool was demonstrated in the 1980s by Clifford Stoll. Stoll, an astronomer working as a systems administrator in the Lawrence Berkeley National Laboratory in California, was asked to investigate a 75 cent discrepancy in the mainframe accounting system. His examination revealed that the accounting anomaly was a trace left by an intruder who had broken into the system, and was also using it as a platform to compromise other systems, including some in the military. Stoll started logging every move made by the intruder, eventually involving a number of US and international law enforcement agencies in an effort to trace and apprehend him. They discovered that he was in Germany, but had difficulty finding an exact location using the telephone call tracing technology of the time. In order to keep him online long enough, Stoll began creating documents related to a fictional "Strategic Defense Initiative", writing a whole suite of convincing documents and letters. The ploy was a success, and led to the arrest of Markus Hess, a West German citizen who broke into computer systems and sold the information to the KGB. Stoll published a paper describing his experiences [16] and expanded it in an engaging and accessible book, "The Cuckoo's Egg" [17], that is highly recommended reading for anyone with an interest in cyber deception.

Research into deception techniques and implementation of honeypots has grown, led by organisations such as the Honeynet Project[1]. Deception is also getting commercial attention now, with a number of companies gaining traction with deception products, including Attivo[2], Countercraft[3], Thinkst[4] and TrapX[5]. Australian cyber company Penten is one of the participants in the work described in this paper. Penten[6] has a deception based data theft detection offering called HoneyTrace[7] that allows clients to create customised fake documents, database entries, credit card numbers, URLs and email addresses. The documents can trigger a beacon when opened, and the URLs and email addresses are monitored for activity. Additionally, active search of social media and the dark web can detect the appearance of any of the artefacts, providing evidence of a breach.

Creating realistic deceptive IT artefacts automatically and at scale remains difficult. In the next section, we describe some key challenges and the research opportunities offered by them.

III. Challenges and Opportunities

The development of cyber deception technology involves creating the deceptive environment, as well as a model of the attacker and ways to measure their perception of the deception. Our approach is to develop rich and believable deceptive environments, populated with deceptive versions of assets that

[1] https://www.honeynet.org
[2] https://www.attivonetworks.com
[3] https://www.countercraftsec.com
[4] https://thinkst.com
[5] https://www.trapx.com
[6] https://www.penten.com
[7] https://honeytrace.io

appear in the IT stack. In addition to creating these artefacts, we also simulate the behaviour of people and their interactions within the system.

Since documents are often sought after by attackers, a particular focus within our group has been on the generation of documents and the simulation of people's interactions with them. The concept of a document honeyfile for intrusion detection was developed by Yuille [18]. A series of papers from the Intrusion Detection Systems Lab at Columbia further explored honeyfile creation and deployment, including some studies of honeyfile characteristics [19]–[23], and Whitham [24] proposed automated topical content creation. Building on this work, our interest in documents has broadened over time from typical office documents to include a range of media that can be found on computing systems, including source code and web/wiki resources. We have also considered the file systems in which generated documents can be found.

Databases are also frequently the target of network intruders, as they are a likely source of high value data such as passwords, payment details and other sensitive personal information. Data breaches of this nature are among the most publicised as they constitute massive privacy breaches and put both individuals and organisations at a variety of risks. There has also been research on creating fake data to populate database honeypots using approaches like rule mining [25] and deep learning with differential privacy [26].

The task of automating the generation of these artefacts has naturally led us to focus primarily on three ML technologies for developing deceptive environments: language models (for creating fake textual content), temporal point processes (for simulating interaction events) and graph neural networks.

The recent development of the Transformer architecture and language models like GPT-2 [27] have radically changed text generation. The new models have enabled realistic text synthesis with freely available pre-trained models. We use both GPT-2 and earlier Recurrent Neural Network (RNN) approaches to sequence generation [28] to create document content. The temporal characteristics of behaviour are modelled using Temporal Point Processes (TPPs), where a number of neural approaches have improved the flexibility and application domain [29]. Artefacts such as file systems and database schema can be modelled as graphs, so we use graph approaches like Graph Recurrent Attention Networks (GRAN) [30] to develop bespoke models.

A. Creating documents

Real world documents are not just text, but can include images, plots and hyperlinks. In addition, our notion of documents includes user data artefacts such as usage history and user files, system files that support the illusion of a live environment, graphics, and other proprietary artefacts like code repositories.

a) Source code: In [31], we address the generation of honeycode - fake software repositories that look like real code when observed through a repository search engine or by command line file content display. In other words, the honeycode need only look like real code when briefly inspected, but does not need to compile. It does, however, have to include a realistic folder structure, file names and mix of file content.

```
function sparse!(d::AbstractMixtureModel, x)
    K = ncomponentwise_logpdf!(Matrix{eltype(x)}(undef, nd, n), one(x))
    logc0 = /x * detach(1)
    logc0 = max(length(IO)*length(J)+1

    for i = 1:(length(Acolptr) <= 1)
        VR = similar(A[1], T, nrows, ncols)
    elseif size(X, 2) == 1
        logX = log(u)
        cm2 /= sqrt(1 - abs2(z)/2) * sqrt(z2)
        alpha3 = max(unsqueezeb, Tuple{Float64})
        $fname(pp, pos, durbin, sort)
            return (false, Int[])
        end
    else
        if !in_single_quotes
            const_prop_profitable(buf) == 0 && return false
            write(buffer, ' ')
        end
    end
    return nothing
    return write_project(env)
end
```

Fig. 1. A code snippet generated by HoneyCode's file content generator. Several patterns emerge that create the illusion of realism including indentations, variable declaration, for-loops and type defining.

Modeling general source code is a challenge because of the rich distribution of languages found within a population of software repositories. The majority of repositories contain a mixture of languages that are either programming or natural (e.g. readme files). Furthermore, these two forms of language can be found in the same file with inline code comments. Modelling programming languages is difficult due to its highly structured nature that requires complex and long range dependencies across blocks of code. We must also consider programming language dependencies across files and hierarchies in the repository structure.

We regard a software repository as a structured amalgamation of three core components: directory structure, filenames and file content. GRAN was specialised to generate trees for modelling file systems, and conditional language generation used to improve the dependencies between different components. For example, we inject filename extensions (e.g. `py`, `txt`) into the file content generator to create consistency between the file types and their respective content. Recurrent Neural Networks trained on 3254 publicly available Julia software repositories from Github are used to generate filenames and file content. A sample of the generated code is provided in Figure 1 along with an overview of the three stage system in Figure 2.

Whilst this work is promising, the file content generator struggles with medium-long range dependencies due to the limited model capacity. This would certainly be improved using large scale Transformers [32] for file content generation due to their ability to model distant dependencies through the use of a global receptive field, at the cost of requiring more training data.

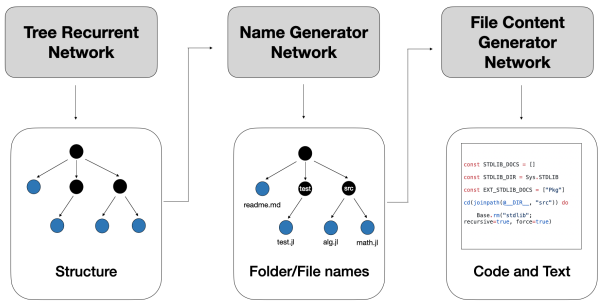

Fig. 2. HoneyCode is composed of three different networks generating three core component, each using a different underlying structure to improve model inductive bias and realism.

b) Websites/Wikis: Our work in [33] explores the use of GPT-2 to generate structured text in wiki-style web formats. Since these documents are written in Markdown, the document structure is encoded with the text. Section headings and other document structural elements are represented by Markdown symbols, which are used to render the documents. Fine-tuning GPT-2 with the Markdown symbols included in the vocabulary successfully enables generation of appropriately structured wiki-style pages. Synthetic articles will consistently begin with an introductory section, followed by a varying number of body sections and subsections, before concluding with references. Occasionally realism-breaking artefacts do emerge in synthetic articles, for example consecutive sections with the same name. Generated pages must be linked to create a realistic wiki, so a GRAN model is successfully trained to generate a page link network. The articles are then represented by the Smoothed Inverse Frequency (SIF) embedding [34] of their content and mapped to nodes in this page link network using a greedy heuristic algorithm. Articles with higher semantic similarity will then be more likely to share an edge in the link network.

The wiki generation process is shown in Figure 3.

c) Document layout: Graphical layouts are ubiquitous across a wide variety of assets found in corporate environments such as documents, posters and presentations. Instead of using human curated templates, recent research [35]–[37] has focused on training deep learning models to generate realistic and authentic layout designs for the purpose of improving graphical editor recommendations. These learning-based approaches are promising for constructing more adaptive and dynamic honeyfiles.

The earliest proposed layout generation approaches were based on variational auto-encoders (VAE) [36], [37] or generative adversarial networks (GAN) [35] frameworks. Experiments indicate that these architectures are frequently subject to posterior collapse or instability in the latent space [38] leading to higher levels of sample degradation. In addition, these methods struggle to exhibit high level of output diversity due to their reliance on uni-modal prediction for object placement.

Our recent work on LayoutMCL [39] demonstrates the feasibility of combining an auto regressive approach with multi

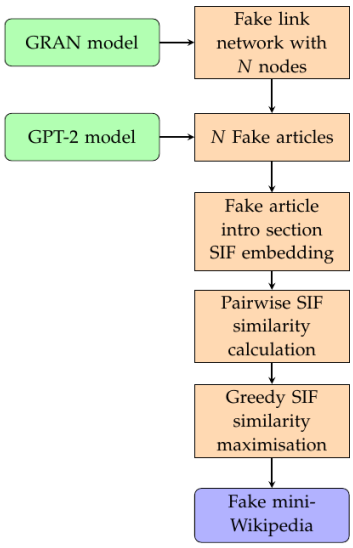

Fig. 3. Wikigen is composed of two different models generating the article link network (GRAN) and the article content (GPT-2), followed by an optimisation procedure to have semantically similar articles be more likely to link to each other.

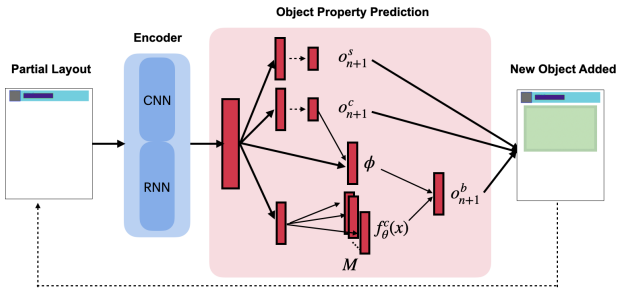

Fig. 4. LayoutMCL is an autoregressive architecture that combines a multimodal encoder and triple head decoder to generate realistic and diverse layouts

choice learning [40]. This unique architecture, shown in Figure 4, generates significantly more stable, realistic and diverse layouts in comparison to its counterparts. This approach is also amenable to incorporating constraints such as a corporate logo or other consistent elements between layouts. These properties make LayoutMCL more suitable for fully automated honeyfile generation.

d) Measuring the enticement of honeyfiles: Looking beyond generation, it is useful to characterise honeypots to guide their creation and deployment. This is particularly true of honeyfiles, where there is a great deal of flexibility in the content, appearance and placement of the deceptions. A number of honeyfile characteristics have been considered [20], [24], including enticement (or enticingness), conspicuousness, believability and realism. We are primarily interested in two metrics: enticement as a measure of how well a honeyfile can attract the attention of an intruder, and realism as a measure

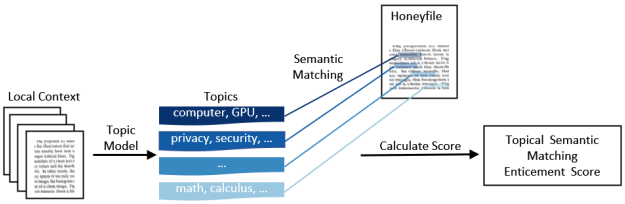

Fig. 5. Visualisation of the Topic Semantic Matching enticement score for deception files.

of how plausible a mimic it is.

It is necessary to construct a model of the adversary in order to meaningfully develop such metrics, and make assumptions about how they will find and interact with the honeyfile. If we assume that an intruder is trying to steal documents from an organisational document repository, we can design honeyfile text with the repository search interface in mind. Honeyfile content, to be enticing, should mimic the topics appearing in the real documents they are protecting so that the honeyfiles will be encountered by someone searching for those topics. This also means accounting for the fact that search engines often show snippets of text containing the search terms in the results, so the fake text should appear realistic and at least locally coherent. Earlier work on enticement [24] measured common word counts as a basis for a metric. Since this approach does not account for paraphrasing, we have introduced a measure called the Topic Semantic Matching (TSM) enticement score [41]. TSM extracts the main topics of a repository, also know as local context, using topic modelling. Next, we calculate the similarity between all the extracted topics and the words in the deception file. This similarity uses a vector space representation of the words in which semantically similar words are close together, so it can detect similarity even when different words with similar meanings are used. A visualisation of the concept of TSM is shown in Figure 5. Experiments show that the TSM measure achieves a higher score when a deception file of a specific theme is compared to a local context of the same theme as shown in Figure 6.

We are also interested in the presence of sensitive information, potentially derived from training or fine-tuning a language model on real documents, appearing in a honeyfile. This problem has been studied in a more general setting associated with classifying document sensitivity automatically. Typical models are based on the word occurrence and corresponding association rules [42], [43], and more recent research [44] experimented with RNNs to detect sensitive sentences.

Recent use of fine-tuning on pre-trained Transformer language models has proven effective on a range of NLP tasks, and we have subsequently tested this idea on sensitive information detection. In [45], we experiment with the fine-tuned Bidirectional Encoder Representations from Transformers (BERT) [46]. This method is graphically represented in Figure 7. Experiments on the Monsanto trial data set show that the fine-tuned BERT model performs better than earlier approaches.

Measuring the realism of a honeyfile is a complex task. There is a considerable research on realistic image [47], [48] and text [49]–[53] generation and, since Transformer models were released, on the perceived realism of generated text [54], [55]. The realism of full documents has not been widely addressed, however. The combination of text, images, layout and format all contribute to the realism of a document, and the context in which it is observed is crucial. If, for example, a document repository search engine returns snapshots of the document cover or page on which the search results appear, the visual attributes will have a significant impact on the whether it is identified as a deception.

We are currently conducting experiments testing perceived enticement and realism with crowd-sourced participants.

B. Simulating deceptive behaviours and interactions

An important aspect of developing realistic deceptive environments is to simulate the various types of interactions seen on real enterprise networks, such as those on networking devices like routers, as well as employee interactions on direct messaging platforms like email and Slack/Teams. In particular, we want to develop generative models to simulate timestamped, directed communications between nodes on these networks. A natural choice for this task is Temporal Point Processes (TPPs), which are generative probabilistic models for event data with timestamps.

Depending on the type of network, communications may be uni-cast or multi-cast. A tool that would be especially handy to have in a cyber deception toolkit is an all-in-one model that has the ability to simulate any type of network communication, eg. uni-cast communications for a deceptive WiFi access point, or multi-cast communications for an email server honeypot.

a) Direct messaging: Within organisations, email has evolved to be much more than just communications. Email platforms are typically integrated with task management tools which encourage the sharing of valuable company assets in the form of attachments in an email. This makes email servers an enticing target for attackers. The simulation of email networks can be split into two subtasks:

1) modeling when and with whom participants communicate, and
2) generating the text to populate simulated conversation threads.

The first task can be tackled with TPPs, and the second task can be achieved with generative language models. A challenging aspect of the first subtask is that emails are *multi-cast* communications, ie. they are directed from one sender to one or more recipients. If we consider a sender who is drafting an email, we can model their choice of whom to include as recipients as a multi-label classification problem.

To the best of our knowledge, the interaction-partitioned topic model (IPTM) of Kim [56] is the only model in the literature that generates message events as well as their textual content. IPTM generates Multi-cast communications using the common multi-label classification approach, which is to model

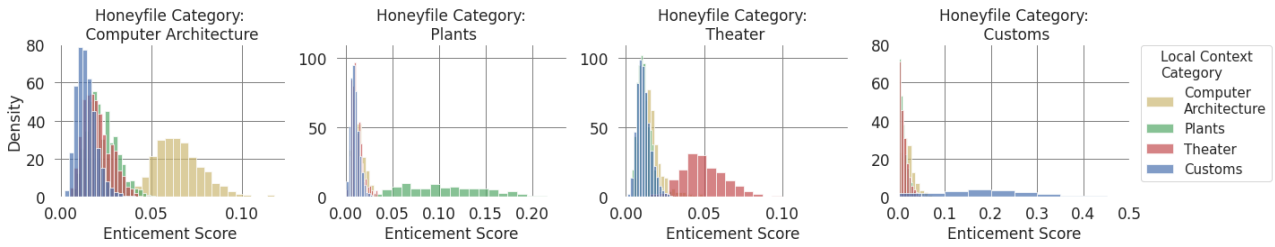

Fig. 6. The distribution of the Topic Semantic Matching enticement score.

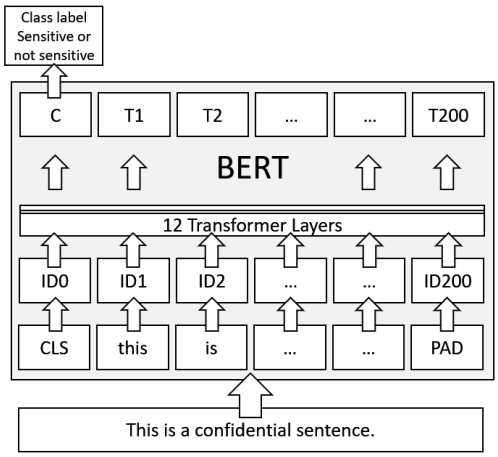

Fig. 7. Visualisation of sensitive information detection with the BERT transformer.

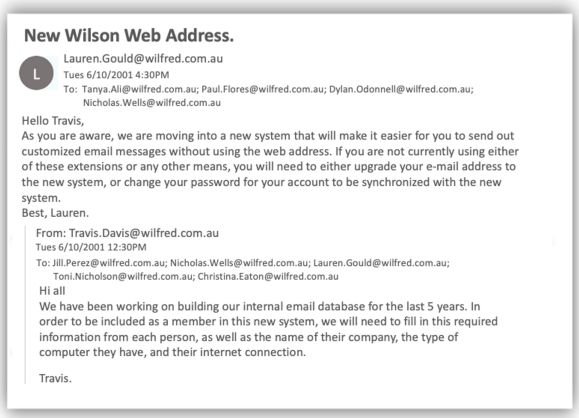

Fig. 8. An example email thread generated from our direct messaging network simulation model.

hyperedges as collections of independent edges, and then use binary classification independently on each edge incident to the sender node to decide whether or not to include that person in the recipient set. This modeling approach presents limitations in generating realistic events, namely it can lead to over or underestimation of event rates for communications with more than one recipient [57], and can generate participant sets that were not seen in the training data. In addition to this, IPTM is only able to generate multi-cast communications, and therefore is not suitable for the all-in-one network communication simulator we seek to create. A further limitation of IPTM for application in cyber deception is that it does not generate human readable conversation text. Instead it uses a topic model to generate a set of word counts for each word in the vocabulary under consideration, which would immediately be identified as a deception by an intruder.

In [58], we develop a direct messaging network simulation model for use in cyber deception. To increase the realism of generated deceptive communications, we introduce the LogNormMix-Net temporal point process, which learns edge weights for each hyperedge (as opposed to each pairwise edge as in IPTM [56]). This is achieved by modeling recipient selection as a multi-class classification problem, instead of multi-label binary classification in IPTM. This recipient selection approach also means that the LogNormMix-Net is able to simulate either uni-cast or multi-cast events, fulfilling the requirement for an all-in-one model that can simulate any type of network communication.

We address the second subtask by applying a fine-tuned GPT-2 model to simulate an office email network where communication threads are coherent and stay on topic, and where each individual will have consistent themes to their communications, that are appropriate to their role within the simulated organisation. An example email and reply generated using the LogNormMix-Net together with our email thread generation model is presented in Figure 8.

As a next step, we plan to utilize the LogNormMix-Net to generate other types of deception interactions, such as simulating WiFi access points.

C. Ongoing and Future Work

We have a number of projects that are still in the early stages of development.

a) Databases: The generation of deceptive relational databases can be addressed with two tasks: firstly the generation of relational schemas, including table and column names, and secondly, populating the tables with conforming data (satisfying the schema design and constraints). To the best of our knowledge, there is no existing work that performs

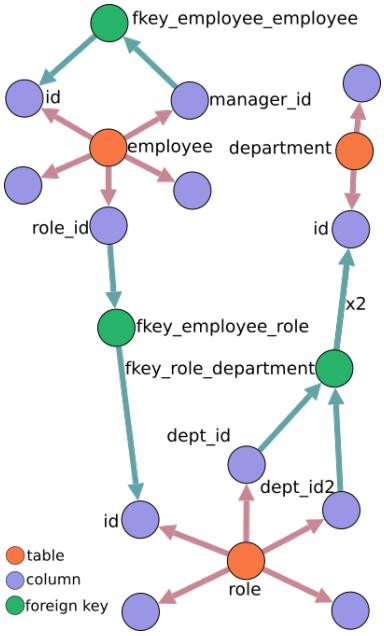

Fig. 9. Graph representation of a relational schema. Some nodes left unlabelled for brevity

both of these tasks to generate completely novel databases. There is work in relational data augmentation [59], however these approaches are not designed to generate consistent and believable data from scratch.

As a first step towards tackling database schema generation, we have extracted a dataset of relational schema [60], as we could not find a suitably sized publicly available collection. Given a suitable schema training dataset, we think there are a couple of potentially viable training approaches. Relational structures naturally lend themselves to a graph representation as shown in Figure 9, where edges link tables that are keyed. One potential approach is therefore to use contemporary graph generation models (such as those we use for code repository and wiki generation). These approaches are designed for homogeneous graph types though, whereas relational schemas are naturally heterogeneous structures, so we believe that they will be better modelled by heterogeneous graphs.

Existing work in heterogeneous graph machine learning is primarily concerned with clustering, classification and edge detection, with successful generative approaches confined to specific domains (e.g. molecule generation) [61]–[64]. Therefore we see an opportunity in the development of an agnostic heterogeneous graph generator.

b) Interpretable generative models: Control over content generation is increasingly important as the need for realism grows. Many generative approaches encode a compressed representation of the model domain called the latent space [65]. Artefacts are generated by sampling from the latent space and *decoding* back into the model domain. This process does not in general, capture interpretable features in the latent space,

Fig. 10. Disentangling colors (rows) and shapes (columns) on shoes data [69], [70]

and so does not allow for fine control over the characteristics of the generated artefacts.

In this project, we study methods of disentangling that enable us to have this control over the generation. For example, consider the images shown in Figure 10. This figure is an output from an extension of VAE/GAN model [66] whose latent variables are shape and colour. Each image is generated by sampling from latent shape and colour variables independently and decoding. We note that changing the shape (variously colour) representation as we move left (variously top) to right (bottom) keeps the colour (shape) content the same, but changes the shoe shape (colour). In representation learning this property is known as a disentanglement; changes in a latent variable of a generative model explicitly changes a distinct aspect of the generated artefact [67].

Our goal is to establish methods to bias generative models to represent features of interest to us in the latent space as latent variables and in a disentangled form. This allows us to alter features of the generated data independently to match a desired distribution. However, disentangled generative models cannot consistently be obtained using unsupervised learning [68], so we focus on semi-supervised methods to disentangle for interpretable features.

c) Graph generation: Observing that graph representations are useful in contexts including file systems, database schema (see § III-C-a), participant users in a communications network and wiki page connection, we're interested in indirect graph generation approaches. Recent work on using deep learning in graph generation can model attributed graphs [64]. While these have performed well, there has been little research into allowing generation conditional on some input that informs the type or structure of the graph. This is of particular importance when the items in the training data are highly diverse, and the model usage would benefit from controlling the generated graph to some extent. To date, only one paper explores using a simple attribute vector for context in generation [71]. Similar to how an image has one or more natural text descriptions, graphs may have a textual

description too, but no work to date incorporates long or complex attributes for conditioning. OpenAI's DALL-E [72] demonstrates impressive text-to-image generation, with a user-supplied text input used to generate an matching image. We are investigating a similar approach for graphs, where a user-supplied text input is used to generate a matching graph.

In particular, generating temporal networks that represent a social network with communications between the (synthetic) participants presents an opportunity for exploring the benefits of text conditioning on generated graphs. The communication dynamics and propagation of topics within social networks is correlated with the topic and type of communication. Allowing a model to learn and depend on a given topic may improve realism significantly.

d) Availability of Datasets: The simulation of rich and realistic environments requires models for a large range of artefacts and activities. One of the biggest limitations we have come across in achieving this goal is the scarcity of publicly available datasets for several of the artefacts or user behaviours we wish to generate.

For some of these tasks, the data is not available for reasons of privacy and security - such as timestamped system logs of user actions. For other tasks, several individual examples can be found on the internet - e.g. database schemas, but tools must be created to locate and extract the individual examples, and then transform them into a standardised collection that is appropriate for ML tasks.

There are therefore two open problems for datasets. The first is the release of new datasets that meet trustworthy and responsible AI requirements, including fairness and privacy. Examples of existing research works in this area include netflow datasets that are collected from real networks and then anonymised and released by the network owner [73]. In the absence of access to real-world data sources, generating synthetic data from a cyber range may also be a possibility [74]. The second open problem is the development of tools that can be used to extract new datasets. Recent examples of extraction-based datasets include WIT: Wikipedia-based Image Text Dataset [75] and CC12M: Conceptual 12M [76].

Some examples of our work in this direction includes:

- Development of a sophisticated chart extraction tool to enable the extraction of a captioned chart dataset.
- Development of the SCHEMADB dataset - a collection of relational schema in both `MySQL` and heterogeneous graph form [60].
- Development of a honeyfile corpus for use in experiments on measuring the enticement of honeyfiles [41].

IV. Discussions and concluding remarks

In this work we have demonstrated various ways to use ML for generating deceptive content, and have highlighted some of the complexities and challenges. Generating realistic content and devising metrics to measure realism remains a challenge, as does finding datasets with which to train models. We recognise that content generation approaches that are privacy preserving, and which prevent language models like GPT from generating toxic content [77], [78] are going to be increasingly important issues that we should pay attention to.

We have also discovered uses for generated content and behaviour outside of deception. Cyber range environments are a rapidly growing application domain for realistic simulations. Ranges have a host of important applications, including security testing and research, security education and capability development [79], all of which rely on content generation to make each environment distinct and convincing. These kinds of *digital twins* are increasingly used in a host of simulation activities [80].

Another area that is evolving is Autonomous Cyber Operations (ACO). There are a number of ACO frameworks, much like cyber ranges, designed for training autonomous attacker and defender agents [81]–[84]. Since simulated ACO environments abstract away information that may be critical to an agent's effectiveness, ACO agents can experience a 'reality gap' between their performance on the simulation environments they are trained on compared to a real network environment [81]. Making ACO environments as realistic as possible is one step to reduce this phenomenon. ACO environments are starting to incorporate features like benign user (gray agent) simulation [82] to make environments richer and more complex, but their behaviours are still quite limited, and several ACO training platform developers have emphasised the importance of, and opportunity for, improvement in the realism of their simulations [81], [83], [84].

The behaviour of an intruder faced with cyber deception is another area of considerable interest. It seems likely that awareness of the possibility that any interaction with a system or artefact could reveal them causes greater caution and evaluation of the risk of each interaction. The behaviour and psychology of the adversary in the presence of cyber deception is increasingly a subject of study [85]–[87]. Even at this early stage, however, it is fair to say that, from the defender's perspective, anything that makes intrusion and theft more costly for the perpetrator should be encouraged.

In closing we should note that, despite extolling the virtues and benefits of deception throughout this paper, we are not advocating it as an alternative to more common security measures. Rather, deception is a complement to existing, perimeter focused security. It acknowledges the risk that, even with the best security, breaches will happen, and provides an additional layer of protection through early detection, adversary intelligence and disruption.

V. Acknowledgements

The work has been supported by the Cyber Security Research Centre Limited whose activities are partially funded by the Australian Government's Cooperative Research Centres Programme. The authors would like to thank the students who participated in the project: Ricard Grace, Alexander Bunn and Adam Green, as well as the CSCRC vacation students Renee Selvey, Duy Khuu and David Liu.

REFERENCES

[1] Gartner. (2017) Gartner forecasts worldwide security and risk management spending to exceed $150 billion in 2021. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2021-05-17-gartner-forecasts-worldwide-security-and-risk-managem

[2] N. E. Weiss and R. S. Miller, "The target and other financial data breaches: Frequently asked questions," in *Congressional Research Service, Prepared for Members and Committees of Congress February*, vol. 4, 2015, p. 2015.

[3] Target Corporation. (2013) Target confirms unauthorized access to payment card data in u.s. stores. [Online]. Available: https://corporate.target.com/press/releases/2013/12/target-confirms-unauthorized-access-to-payment-car

[4] B. Krebs. (2014) The target breach, by the numbers. [Online]. Available: https://krebsonsecurity.com/2014/05/the-target-breach-by-the-numbers/

[5] Reuters. (2017) Target settles 2013 hacked customer data breach for $ 18.5 million. [Online]. Available: www.nbcnews.com/business/business-news/target-settles-2013-hacked-customer-data-breach-18-5-million-n764031

[6] M. N. Harrell, "Synergistic security: A work system case study of the target breach," *Journal of Cybersecurity Education, Research and Practice*, vol. 2017, no. 2, p. 4, 2017.

[7] M. Plachkinova and C. Maurer, "Security breach at target," *Journal of Information Systems Education*, vol. 29, no. 1, pp. 11–20, 2018.

[8] M. J. Schwartz. (2014) Target ignored data breach alarms. [Online]. Available: https://www.darkreading.com/attacks-and-breaches/target-ignored-data-breach-alarms/d/d-id/1127712

[9] Ponemon Institute, "2021 cost of a data breach study: Global overview," June 2021.

[10] L. Spitzner, "Honeypots: Catching the insider threat," in *19th Annu. Computer Security Applicat. Conf., Proc.* IEEE, 2003, pp. 170–179.

[11] D. Fraunholz and H. D. Schotten, "Defending web servers with feints, distraction and obfuscation," in *Proc. Int. Conf. Computing, Networking and Communications*. IEEE, 2018, pp. 21–25.

[12] M. Stevens, *Cheats and deceits: how animals and plants exploit and mislead*. Oxford University Press, 2016.

[13] T. J. Smith, "Overlord/bodyguard: Intelligence failure through adversary deception," *International Journal of Intelligence and CounterIntelligence*, vol. 27, no. 3, pp. 550–568, 2014.

[14] J. B. Bell, "Toward a theory of deception," *Int. J. of Intell. and CounterIntell.*, vol. 16, no. 2, pp. 244–279, 2003.

[15] B. Whaley, "Toward a general theory of deception," *J. of Strategic Stud.*, vol. 5, no. 1, pp. 178–192, 1982.

[16] C. Stoll, "Stalking the wily hacker," *Communications of the ACM*, vol. 31, no. 5, pp. 484–497, 1988.

[17] ——, *The Cuckoo's Egg: Tracking a Spy through the Maze of Computer Espionage*. Simon and Schuster, 2005.

[18] J. Yuill, M. Zappe, D. Denning, and F. Feer, "Honeyfiles: deceptive files for intrusion detection," in *Proc. the 5th Annu. SMC Information Assurance Workshop*. IEEE, 2004, pp. 116–122.

[19] J. Voris, N. Boggs, and S. J. Stolfo, "Lost in translation: Improving decoy documents via automated translation," in *Security and Privacy Workshops, IEEE Symp. on*, 2012, pp. 129–133.

[20] J. Voris, J. Jermyn, A. D. Keromytis, and S. J. Stolfo, "Bait and snitch: Defending computer systems with decoys," in *Proc. the cyber infrastructure protection Conf., Strategic Stud. Institute, September*, 2013.

[21] J. Voris, J. Jermyn, N. Boggs, and S. Stolfo, "Fox in the trap: thwarting masqueraders via automated decoy document deployment," in *Proc. the 8th European Workshop on System Security*. ACM, 2015, p. 3.

[22] B. M. Bowen, S. Hershkop, A. D. Keromytis, and S. J. Stolfo, "Baiting inside attackers using decoy documents." in *SecureComm*, vol. 19. Springer, 2009, pp. 51–70.

[23] M. B. Salem and S. J. Stolfo, "Decoy document deployment for effective masquerade attack detection," in *Int. Conf. Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 2011, pp. 35–54.

[24] B. Whitham, "Automating the generation of enticing text content for high-interaction honeyfiles," in *Proc. the 50th Hawaii Int. Conf. System Sciences*, 2017.

[25] M. Bercovitch, M. Renford, L. Hasson, A. Shabtai, L. Rokach, and Y. Elovici, "Honeygen: An automated honeytokens generator," in *Proceedings of 2011 IEEE International Conference on Intelligence and Security Informatics*. IEEE, 2011, pp. 131–136.

[26] N. C. Abay, C. G. Akcora, Y. Zhou, M. Kantarcioglu, and B. Thuraisingham, "Using deep learning to generate relational honeydata," in *Autonomous Cyber Deception*. Springer, 2019, pp. 3–19.

[27] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[28] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[29] O. Shchur, A. C. Türkmen, T. Januschowski, and S. Günnemann, "Neural temporal point processes: A review," *CoRR*, vol. abs/2104.03528, 2021. [Online]. Available: https://arxiv.org/abs/2104.03528

[30] R. Liao, Y. Li, Y. Song, S. Wang, W. Hamilton, D. K. Duvenaud, R. Urtasun, and R. Zemel, "Efficient graph generation with graph recurrent attention networks," in *Advances in Neural Information Processing Systems*, 2019, pp. 4257–4267.

[31] D. Nguyen, D. Liebowitz, S. Nepal, and S. Kanhere, "Honeycode: Automating deceptive software repositories with deep generative models," in *Proc. the 54th Hawaii Int. Conf. Syst. Sci.*, 2021, p. 6945.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[33] M. Longland, "Generating fake websites: Wikigen," Australian National University, Tech. Rep., 2020. [Online]. Available: https://github.com/longland-m/wikigen

[34] S. Arora, Y. Liang, and T. Ma, "A simple but tough-to-beat baseline for sentence embeddings," in *International conference on learning representations*, 2017.

[35] J. Li, J. Yang, A. Hertzmann, J. Zhang, and T. Xu, "Layoutgan: Generating graphic layouts with wireframe discriminators," *arXiv preprint arXiv:1901.06767*, 2019.

[36] A. A. Jyothi, T. Durand, J. He, L. Sigal, and G. Mori, "Layoutvae: Stochastic scene layout generation from a label set," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9895–9904.

[37] H.-Y. Lee, L. Jiang, I. Essa, P. B. Le, H. Gong, M.-H. Yang, and W. Yang, "Neural design network: Graphic layout generation with constraints," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*. Springer, 2020, pp. 491–506.

[38] K. Kikuchi, E. Simo-Serra, M. Otani, and K. Yamaguchi, "Constrained graphic layout generation via latent optimization," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 88–96.

[39] D. D. Nguyen, S. Nepal, and S. S. Kanhere, "Diverse multimedia layout generation with multi choice learning," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 218–226.

[40] A. Guzman-Rivera, D. Batra, and P. Kohli, "Multiple choice learning: Learning to produce multiple structured outputs." in *NIPS*, vol. 1, no. 2. Citeseer, 2012, p. 3.

[41] R. Timmer, D. Liebowitz, S. Nepal, and S. Kanhere, "Tsm: Measuring the enticement of honeyfiles with natural language processing," in *Proc. the 55th Hawaii Int. Conf. Syst. Sci.*, 2022.

[42] R. Chow, P. Golle, and J. Staddon, "Detecting privacy leaks using corpus-based association rules," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 893–901.

[43] D. Sánchez and M. Batet, "C-sanitized: A privacy model for document redaction and sanitization," *Journal of the Association for Information Science and Technology*, vol. 67, no. 1, pp. 148–163, 2016.

[44] J. Neerbek, I. Assent, and P. Dolog, "Detecting complex sensitive information via phrase structure in recursive neural networks," in *Pacific-Asia Conf. Knowledge Discovery and Data Mining*. Springer, 2018, pp. 373–385.

[45] R. Timmer, D. Liebowitz, S. Nepal, and S. Kanhere, "Can pre-trained transformers be used in detecting complex sensitive sentences? - a monsanto case study," in *Proc. the 3rd IEEE International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications*, 2021.

[46] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[47] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[48] A. v. d. Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu, "Conditional image generation with pixelcnn decoders," *arXiv preprint arXiv:1606.05328*, 2016.

[49] W. S. Cho, P. Zhang, Y. Zhang, X. Li, M. Galley, C. Brockett, M. Wang, and J. Gao, "Towards coherent and cohesive long-form text generation," *arXiv preprint arXiv:1811.00511*, 2018.

[50] A. Bakhtin, S. Gross, M. Ott, Y. Deng, M. Ranzato, and A. Szlam, "Real or fake? learning to discriminate machine from human generated text," *arXiv preprint arXiv:1906.03351*, 2019.

[51] J. Bullock and M. Luengo-Oroz, "Automated speech generation from un general assembly statements: Mapping risks in ai generated texts," *arXiv preprint arXiv:1906.01946*, 2019.

[52] P. Karuna, H. Purohit, R. Ganesan, and S. Jajodia, "Generating hard to comprehend fake documents for defensive cyber deception," *IEEE Intell. Syst.*, vol. 33, no. 5, pp. 16–25, 2018.

[53] J. Johnson, A. Gupta, and L. Fei-Fei, "Image generation from scene graphs," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1219–1228.

[54] E. Clark, T. August, S. Serrano, N. Haduong, S. Gururangan, and N. A. Smith, "All that's' human'is not gold: Evaluating human evaluation of generated text," *arXiv preprint arXiv:2107.00061*, 2021.

[55] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[56] B. Kim, "Latent modeling of dynamic social networks," Ph.D. dissertation, The Pennsylvania State University, 8 2018.

[57] P. S. Chodrow, "Configuration models of random hypergraphs," *Journal of Complex Networks*, vol. 8, no. 3, p. cnaa018, 2020.

[58] K. Moore, C. J. Christopher, D. Liebowitz, S. Nepal, and R. Selvey, "Modelling direct messaging networks with multiple recipients for cyber deception," *arXiv preprint arXiv:2111.11932*, 2021.

[59] N. Patki, R. Wedge, and K. Veeramachaneni, "The synthetic data vault," in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2016, pp. 399–410.

[60] C. J. Christopher, K. Moore, and D. Liebowitz, "SCHEMADB: Structures in relational datasets," *arXiv preprint arXiv:2111.12835*, 2021.

[61] X. Wang, D. Bo, C. Shi, S. Fan, Y. Ye, and P. S. Yu, "A survey on heterogeneous graph embedding: methods, techniques, applications and sources," *arXiv preprint arXiv:2011.14867*, 2020.

[62] X. Guo and L. Zhao, "A systematic survey on deep generative models for graph generation," *arXiv preprint arXiv:2007.06686*, 2020.

[63] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *arXiv preprint arXiv:1901.00596*, 2019.

[64] F. Faez, Y. Ommi, M. S. Baghshah, and H. R. Rabiee, "Deep graph generators: A survey," *IEEE Access*, vol. 9, pp. 106 675–106 702, 2021.

[65] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *arXiv preprint arXiv:1906.02691*, 2019.

[66] A. Srivastava, L. Valkov, C. Russell, M. U. Gutmann, and C. Sutton, "Veegan: Reducing mode collapse in gans using implicit variational learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 3310–3320.

[67] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[68] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *Int. Conf. machine learning*. PMLR, 2019, pp. 4114–4124.

[69] A. Yu and K. Grauman, "Fine-grained visual comparisons with local learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 192–199.

[70] ——, "Semantic jitter: Dense supervision for visual comparisons via synthetic images," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5570–5579.

[71] C. Yang, P. Zhuang, W. Shi, A. Luu, and P. Li, "Conditional structure generation through graph variational generative adversarial nets." in *NeurIPS*, 2019, pp. 1338–1349.

[72] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," *arXiv preprint arXiv:2102.12092*, 2021.

[73] M. J. M. Turcotte, A. D. Kent, and C. Hash, *Unified Host and Network Data Set*. World Scientific, 11 2018, ch. Chapter 1, pp. 1–22. [Online]. Available: https://www.worldscientific.com/doi/abs/10.1142/9781786345646_001

[74] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, 2015, pp. 1–6.

[75] K. Srinivasan, K. Raman, J. Chen, M. Bendersky, and M. Najork, "Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning," 2021.

[76] S. Changpinyo, P. Sharma, N. Ding, and R. Soricut, "Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts," 2021.

[77] I. Solaiman and C. Dennison, "Process for adapting language models to society (palms) with values-targeted datasets," 2021.

[78] A. Abid, M. Farooqi, and J. Zou, "Persistent anti-muslim bias in large language models," *arXiv preprint arXiv:2101.05783*, 2021.

[79] E. C. S. O. (ECSO). (2020) Understanding cyber ranges: From hype to reality. ECS Brussels, Belgium. [Online]. Available: https://www.ecs-org.eu/documents/uploads/understanding-cyber-ranges-from-hype-to-reality.pdf

[80] D. Holmes, M. Papathanasaki, L. Maglaras, M. A. Ferrag, S. Nepal, and H. Janicke, "Digital twins and cyber security–solution or challenge?" in *2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*. IEEE, 2021, pp. 1–8.

[81] M. Standen, M. Lucas, D. Bowman, T. J. Richer, J. Kim, and D. Marriott, "Cyborg: A gym for the development of autonomous cyber agents," *arXiv preprint arXiv:2108.09118*, 2021.

[82] A. Molina-Markham, C. Miniter, B. Powell, and A. Ridley, "Network environment design for autonomous cyberdefense," *arXiv preprint arXiv:2103.07583*, 2021.

[83] M. D. R. Team *et al.*, "Cyberbattlesim," 2021.

[84] J. Schwartz and H. Kurniawati, "Autonomous penetration testing using reinforcement learning," *arXiv preprint arXiv:1905.05965*, 2019.

[85] K. Ferguson-Walter, T. Shade, A. Rogers, M. C. S. Trumbo, K. S. Nauer, K. M. Divis, A. Jones, A. Combs, and R. G. Abbott, "The tularosa study: An experimental design and implementation to quantify the effectiveness of cyber deception." Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), Tech. Rep., 2018.

[86] K. J. Ferguson-Walter, M. M. Major, C. K. Johnson, and D. H. Muhleman, "Examining the efficacy of decoy-based and psychological cyber deception," in *30th {USENIX} Security Symposium ({USENIX} Security 21)*, 2021.

[87] D. Ashenden, R. Black, I. Reid, and S. Henderson, "Design thinking for cyber deception," in *Proceedings of the 54th Hawaii International Conference on System Sciences*, 2021, p. 1958.