Knowledge Infusion in Privacy Preserving Data Generation

Anantaa Kotal
C.S.E.E. Dept.
University of Maryland,
Baltimore County
anantak1@umbc.edu

Nilanjana Das
C.S.E.E. Dept.
University of Maryland,
Baltimore County
ndas2@umbc.edu

Anupam Joshi
C.S.E.E. Dept.
University of Maryland,
Baltimore County
joshi@cs.umbc.edu

ABSTRACT

Security monitoring is crucial for maintaining a strong IT infrastructure by protecting against emerging threats, identifying vulnerabilities, and detecting potential points of failure. It involves deploying advanced tools to continuously monitor networks, systems, and configurations. However, organizations face challenges in adapting modern techniques like Machine Learning (ML) due to privacy and security risks associated with sharing internal data. Compliance with regulations like GDPR further complicates data sharing. To promote external knowledge sharing, a secure and privacy-preserving method for organizations to share data is necessary. Privacy-preserving data generation involves creating new data that maintains privacy while preserving key characteristics and properties of the original data so that it is still useful in creating downstream models of attacks. Generative models, such as Generative Adversarial Networks (GAN), have been proposed as a solution for privacy preserving synthetic data generation. However, standard GANs are limited in their capabilities to generate realistic system data. System data have inherent constraints e.g., the list of legitimate I.P. addresses and port numbers are limited, and protocols dictate a valid sequence of network events. Standard generative models do not account for such constraints and do not utilize domain knowledge in their generation process. Additionally, they are limited by the attribute values present in the training data. This poses a major privacy risk, as sensitive discrete attribute values are repeated by GANs. To address these limitations, we propose a novel model for Knowledge Infused Privacy Preserving Data Generation. A privacy preserving Generative Adversarial Network (GAN) is trained on system data for generating synthetic datasets that can replace original data for downstream tasks while protecting sensitive data. Knowledge from domain specific knowledge graphs is used to guide the data generation process, check for the validity of generated values and enrich the dataset by diversifying the values of attributes. We specifically demonstrate this model by synthesizing network data captured by the network capture tool, Wireshark. We establish that the synthetic dataset holds up to the constraints of the network specific datasets and can replace the original dataset in downstream tasks.

CCS CONCEPTS

• **Security and privacy** → **Domain-specific security and privacy architectures**; • **Computing methodologies** → *Reasoning about belief and knowledge*; Neural networks.

KEYWORDS

GANs, Knowledge Graphs, synthetic data, privacy, security

ACM Reference Format:
Anantaa Kotal, Nilanjana Das, and Anupam Joshi. 2023. Knowledge Infusion in Privacy Preserving Data Generation. In *M. Gaur, E. Tsamoura, S. Sreedharan, S. Mittal, Proceedings of the Third ACM SIGKDD Workshop on Knowledge-infused Learning (KDD KiL 2023). Long Beach, California, USA, August 6, 2023. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).* ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Security monitoring systems play a vital role in maintaining a robust IT infrastructure within an organization. They help protect against emerging threats and vulnerabilities in software, operating systems, and applications, and identification of potential points of failure in existing systems [5, 13, 15, 17, 18]. Security monitoring involves deploying advanced tools and technologies that continuously monitor the organization's IT infrastructure, networks, and systems. This monitoring allows organizations to detect and respond to security incidents promptly. Timely detection enables them to initiate incident response procedures, minimizing the impact of security breaches and reducing potential financial and reputational damage. By analyzing network traffic, system configuration data, and endpoint data, organizations can identify anomalies, suspicious activities, or indicators of compromise that may indicate a cyber attack in progress. Hence, organizations dedicate considerable resources to collecting such data in their IT system.

As the frequency of cyber threats continues to increase, significant advancements have been made in cybersecurity tools focused on threat detection and vulnerability management. Particularly, the integration of Machine Learning (ML) models have played a pivotal role in enhancing these capabilities [7–9, 19, 22]. However, organizations are limited in their ability to adopt these modern techniques as it requires them to share their internal data for learning. System data often contains personally identifiable information (PII), confidential business data, trade secrets, or other sensitive data that cannot be shared due to regulatory, legal, or business issues. Sharing such data externally can introduce several privacy and security risks. Additionally, there are specific policy regulations, such as the GDPR in the EU, that impose obligations on organizations to ensure data privacy and security, which can make sharing system data a complex and legally sensitive matter. To encourage organizations to share their knowledge externally, both to make their security infrastructure more robust and to aid in cybersecurity research, there are two approaches. One is federated learning[2] where models are trained on partial data within organizations and then joined together. An alternate approach is to provide a secure and confidential way for organizations to share their data.

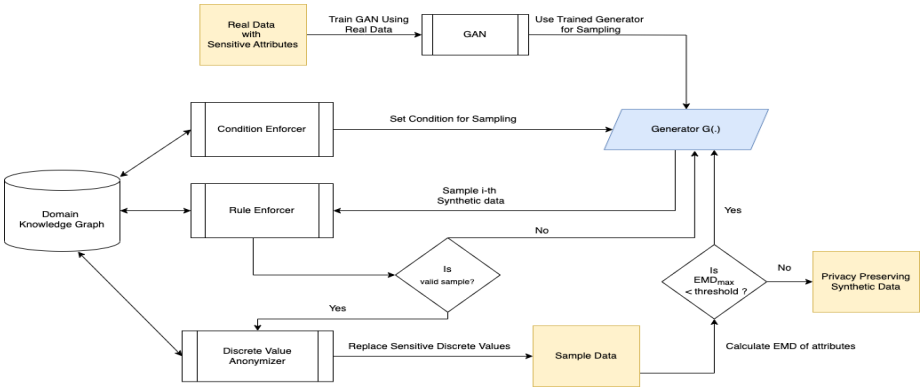

Figure 1: Framework for Knowledge Infused Privacy Preserving Data Generation

Privacy-preserving data generation refers to the process of creating new *synthetic* data that maintains privacy while retaining useful characteristics and statistical properties of the original data. This technique allows organizations or researchers to share or analyze data without directly exposing sensitive information. One of the approaches to generating synthetic data that bears a close resemblance to original data for any analytical task is using generative adversarial networks (GANs). GANs are generative deep learning models that learn the underlying distribution of a training dataset in a two-player min-max optimization model. GANs are often used for synthetic data generation and translation in image and text data [4, 23, 26]. Conditional GANs (CGAN) have been shown to accurately learn the underlying distribution of tabular data, like system monitoring data, that contain a mix of discrete and continuous variables [3, 12, 24, 25]. Synthetic data generated using privacy preserving versions of GAN have been shown to replace real data for statistical and analytical purposes while protecting sensitive information.

However, standard GANs are limited in capabilities to generate realistic system data, like network activity data, that are used in security monitoring tools. For example, in a typical network data capture, a limited set of IP addresses are observed. CGAN for tabular data is limited to repeating observed values of discrete attributes. Hence, even in the synthetic dataset, only the observed values of IP addresses can appear. However, this is a privacy risk since it reveals what addresses are communicating within an organization. Additionally, system data typically contain inherent constraints, e.g., fixed sequence of events in the network activity, fixed range of IP addresses, valid set of port numbers, etc. However, standard GANs do not enforce these constraints. The result is unrealistic generated data that can easily be discerned from the original data and may mislead downstream tasks such as classifiers that are trying to separate legitimate traffic from attacks. We propose infusing knowledge and reason in the generation process of privacy preserving GANs to enrich the synthetic dataset. Information like valid range of attributes can be easily captured and represented in domain specific Knowledge Graphs (KG). We can mine this knowledge from domain specific Knowledge Graphs (KG) to control the data generation process in a GAN.

In this paper, we propose a novel framework for a knowledge infusion in privacy preserving data generation for system data. We show how knowledge from domain specific Knowledge Graphs can be induced in the sampling step of GANs to create realistic synthetic datasets, that can replace original data for downstream tasks. We have demonstrated the use of our framework for the generation of Network Activity Data. For inducting the domain knowledge, we have developed a novel ontology for network data capture that guides the valid sequence of events in a network activity. We demonstrate our proposed framework by training Network Activity Data in a live system through Wireshark and training a GAN model on the captured data and using the network data ontology to guide the generation process. We show the validity of the generated dataset by verifying with a human annotator, using the generated dataset to replace the original dataset in downstream tasks with minimal loss, and showing that the generated dataset is resistant to privacy attacks.

2 PROPOSED FRAMEWORK

Our proposed model uses knowledge infusion from a domain-specific Knowledge Graph to generate enriched synthetic datasets that preserve privacy while still being able to replace original data for downstream tasks. A privacy preserving Generative Adversarial Network (GAN) is used to learn the underlying distribution of the training data. Knowledge from domain specific knowledge graphs is used to guide the data generation process, check for the validity of generated values and enrich the dataset by diversifying the values of attributes. The overall architecture for our proposed framework is illustrated in Figure 1.

2.1 Privacy Preserving Data Generation

Generative adversarial Networks (GANs) are a generative ML framework that attempts to learn the underlying distribution of the training set in an adversarial setting. A typical GAN framework consists of a generative model G that approximates the training data distribution, and a discriminative model D that differentiates between a sample from the original distribution versus a sample generated from G. The training procedure for G is to maximize the probability of D making a mistake. With each iteration, the generator gets better at approximation and sampling from the underlying distribution

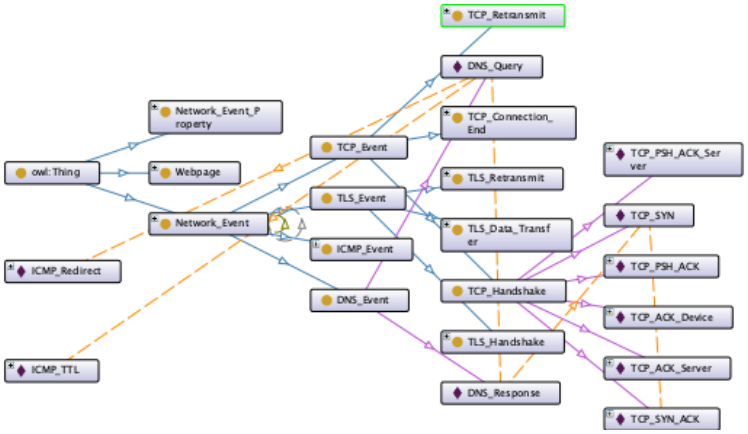

Figure 2: Ontology for Network Activity Capture

of the training data. The idea behind GAN can be formulated as a two-player min-max game with value function $V(G, D)$:

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

In the case of system data, the data is usually tabular i.e. a mix of discrete and continuous values. To account for this, we need specialized versions of GAN that can accurately replicate system data that is collected from our system. In Conditional Generative Adversarial Nets (CGAN) [14], the generator is trained by conditioning the model on a constraint. It is thus possible to direct the data generation process.

Additionally, we have to ensure that the data generated using GAN preserves privacy. The PriveTab generator [12], models tabular data in three key steps (1) Mode-specific normalization, (2) Conditional Generator, and (3) Training by sampling. Additionally, the model ensures that the synthetic dataset is **t-close** to the original dataset to enforce privacy. For this, the Earth Mover's distance (EMD) of the distribution of features in the synthetic is calculated w.r.t. the original dataset. We train a GAN for privacy preserving synthetic data generation using the method described in Privetab.

2.2 Knowledge Graphs and Network Activity Ontology

Knowledge Graphs (KG) are graph-structured data models used for knowledge representation and reasoning. The information or knowledge in a KG is structured as semantic triples of subject, predicate, and object. Here subjects and objects are nodes or entities in the graph and predicate is the functional relation between them. Due to the graph like structure of KG, large quantities of information can be stored in the KG and the network can be extended with new knowledge as needed. The KG also has powerful reasoning capabilities that allow us to put constraints on the entities and infer new knowledge. For example, we could say that source IPs must not be from within a subnet, or must originate from a specific external CIDR range. Due to its strong knowledge representation and reasoning abilities, we can use KG to augment our data generation process.

```
SELECT ?event2
WHERE {
        ?event1 nm:Followed_By ?event2
FILTER (?event1=
        http://www.semanticweb.org/****/ontologies/2023/1/networkontology#
        )
}
```

Figure 3: SPARQL Query to Retrieve Subsequent Network Event

```
SELECT ?portNumber
WHERE {
 BIND(1234 AS ?portNumber)  # Replace 1234 with the value you want to check
 FILTER(?portNumber >= 0 && ?portNumber <= 65535)
}
```

Figure 4: SPARQL Query to Check Validity of Port Number

For our use case, we have designed a Network Activity ontology as an extension of Unified Cyber Ontology (UCO)[21] to maintain the sequence of events as interpreted by network capture tools like Wireshark. Figure 2, Our ontology introduces three main classes: "Network_Event", "Network_Event_Property" and "Webpage". "Network_Event" describes individual network events in network traffic and each event has properties like Source and Destination IP, Port Address and associated Network Protocol that are described in the "Network_Event_Property". Each "Network_Event" is subdivided into 4 classes based on their protocol: "DNS_Event", "TCP_Event", "ICMP_Event" and "TLS_Event". Each event in Network Event has a temporal sequence. For instance, a DNS query should be followed by a DNS response and DNS Response can only come after a DNS Query. For this purpose, the "Network_Event" has relationships 'Followed_By' and 'Preceded_By' to another "Network_Event" which dictates the ordering of events in network activity data. We use this KG to guide our data generation process. We train generative AI models on our system data.

```
SELECT ?ipAddress
WHERE {
 ?ip uco-core:observableData ?ipAddress .
 ?ip a uco-observable:IPv4Address .
}
```

Figure 5: SPARQL Query to Retrieve Valid IP addresses

2.3 Knowledge Guided Data Generation

In reality, the discrete attribute can take on a larger range of values. Additionally, there are strict rules in a system that have to be observed. One such example is that in a network activity, certain network packets have to be followed by others to establish a legitimate connection. Generative models can not adhere to these rules unless specifically constrained.

We can sample the trained generative model to synthesize a new dataset. However, the sampling process is not constrained and this can lead to fallacies in the generated dataset. We use the knowledge in our KG to guide us in this process. We enrich the sampling process from the trained generative model through 3 key steps:

(1) **Condition Enforcer:** Prior knowledge can be induced in the generation process through Knowledge Graph Query. Information about previously generated samples can be used in conjunction with data from the Knowledge graph to set condition for the conditional generator. For example, the Network Activity Ontology contains information about the temporal sequence of events in Network Data Capture. We can query the knowledge graph to find out which event logically follows the last generated event from the GAN. An example of such query is given in Figure 3. The result of the query can be added as the conditional input to the GAN.

(2) **Rule Enforcer:** In case of tabular data, CGANs treat each attribute as either discrete or continuous. It is not able to model conditionally continuous variables or variables that are continuous within a range. For example, port addresses are numbered values that are valid within a range. To enforce this, we use a rule enforcer that checks the validity of the generated data and rejects data that violates a constraint. For example, the knowledge graph can be queried with a generated port number to check whether the port number is valid. An example of such query is provided in Figure 4.

(3) **Discrete Value Anonymizer:** One of the limitations of data generation using GANs is that the models can only learn from the data it has seen. This means it can only repeat the discrete values from the training set which is limited by the values we have observed during testing in our digital twin. This is a privacy risk as actual observed values of attributes can be sensitive. In the case of network activity data for example, we can only use the IP addresses with observed network activity. Using only that data that we have observed in our system, the generative model has a limited view of possible IP addresses. A discrete value anonymizer queries the Knowledge Graph for possible values of discrete attributes such as **IP address, port number, domain address** etc. Thus, it decreases the likelihood of adversaries identifying actual observed values from the synthetic data. An example of such query is provided in Figure 5.

3 TESTING AND VALIDATION

The objective of our framework is to produce realistic and privacy preserving datasets that can replace sensitive system data for downstream tasks. To test our proposed framework, we use it to generate synthetic data fro Network Activity Data. Network Activity Data includes sensitive information such as source and destination address, and data contained in packets. We use our framework to synthesize a completely fake dataset of Network Activity data which while hiding actual sensitive information is still useful for other tasks. For this collect our own dataset of Network Activity data from a system of connected IT and IoT devices through the network capture tool Wireshark. We use 3 validation techniques to show that the synthetic data is privacy preserving, retains relevant information and can replace the original dataset with minimal loss in accuracy.

3.1 Data Collection

In our experimental setup we have multiple IT and IoT devices that are connected in our network, including a Blink camera, a smart plug with lamp and a motion sensor. We monitor the network traffic in our system of IoT devices using network capture tools like Wireshark. We filtered the data with the IP addresses of the IoT devices to comprehend the communications that the devices usually take part in. The network traffic data that we collected consists of features like Source IP address, Destination IP Address, Source Port, Destination Port, Protocol, etc. In case of the camera, data was collected particularly when motion was detected in front of it. The smart plug with lamp data consists of activity like turning on/off the lamp. The motion sensor data consists of communications that the tag manager makes. The tag manager is connected to the tag or the sensor.

3.2 Validation

We use our proposed framework to synthesize Network Activity data by training on our collected data. We use the Network Activity Ontology as described in Section 2.2 to guide our data generation process and synthesize our generated dataset. The synthetic dataset is validated for logical consistency using a human expert. We also check the accuracy of the synthetic dataset in downstream tasks and privacy preserving properties against an adversary.

3.2.1 Human Validation. We asked a human expert to analyse the synthetic dataset and manually annotate observed sequences of events. In a valid network activity dataset, network events follow a valid sequence of events and packets. For example, a TLS Handshake is always preceded by a TCP Handshake. Additionally in a TCP handshake, a SYN-ACK packet should be preceded by a SYN packet. In our original data, first, a DNS request is made from 192.168.1.122 (camera) to 192.168.1.1 (router gateway) and a response is received from the gateway to the camera. In our synthetic data, the human expert observed that a DNS request - response first takes place between 192.168.1.168 (camera) and 192.168.1.1 (router's gateway). This sequence of events is then followed by a TCP 3 way handshake process, both in our original and synthetic data. Our synthetic data

Seq. No.	Source	Source Port	Destination	Destination Port	Protocol	Length	Info	Annotated Events
43869	192.168.1.68	54176	192.168.1.1	53	DNS	60	Standard query 0xbc55 A VS-cam.u038.immedia-semi.com	DNS Query
...								
45863	192.168.1.68	63542	54.164.231.176	443	TCP	590	63542 >443 [SYN] ,Seq=1804 Win=26883 Len=0 MSS=0	TCP Handshake
46022	54.164.231.176	443	192.168.1.68	63542	TCP	1467	443 >63542 [SYN, ACK] ,Seq=1804 Ack=93857 Win=33580 Len=277 MSS=0	
46026	192.168.1.68	63542	54.164.231.176	443	TCP	580	63542 >443 [ACK] ,Seq=599 Ack=1804 Win=33580 Len=0	
46027	192.168.1.68	63542	54.164.231.176	443	TLSv1.2	157	Client Hello	TLS Handshake
46189	192.168.1.68	63542	54.164.231.176	443	TLSv1.2	54	Application Data	Data Transfer
...								
48405	54.164.231.176	443	192.168.1.68	63542	TCP	70	"443 >63542 [FIN, ACK] "Seq=1804 Ack=1804 Win=33580 Len=0	TCP Connection End
50281	54.164.231.176	443	192.168.1.68	63542	TCP	1467	443 >63542 [RST] "Seq=1804 Win=33580 Len=0	

Table 1: Samples from Generated Network Activity Data with human annotation

Classifier Model	Accuracy on Original Data	Accuracy on Generated Data
Logistic Regression	0.98	0.96
Decision Tree	0.98	0.97
Random Forest	0.98	0.98
XGBoost	0.99	0.98
Average:	0.98	0.97

Table 2: Comparison of Anomaly detection Classifier models trained on Original vs Generated Data

also has the TLS events taking place between the camera and the server. Once the data is transferred, 54.164.231.176 sends a FIN-ACK packet to 192.168.1.68 (camera) to terminate the connection. Samples from our synthetic Network Activity dataset along with the human annotation of network events are provided in Table 1.

3.2.2 Accuracy in Downstream tasks. Network activity datasets are commonly used to identify network events. For example, in the network activity data collected from our IoT system, ML models are used to automatically detect network activity corresponding movement in front of a camera and any data being transferred from the camera due to the movement. Supervised ML models are trained on annotated network activity dataset to label network activities to identify such network events. Any ML model trained on the original data should produce similar results to an ML model trained on a dataset generated through our framework. To prove this, we train ML models on both original and synthetic datasets and compare their accuracy against a test dataset retained from the original dataset. The results of our comparison are provided in Table 2. The average accuracy of ML models trained on the original dataset is **0.98**. In comparison, the average accuracy of ML models trained on the generated dataset is **0.97**. Thus there is minimal loss in accuracy for ML models trained on generated data.

3.2.3 Resistence against Privacy Attack. A common privacy breach is identifying original datapoints from a shared dataset. Given a shared synthetic dataset, adversaries try to identify specific datapoints that also belong to the original dataset. The identification of membership of a datapoint in an original, sensitive dataset is a privacy risk and we show here that our model is resistant to such attacks. We test our model with an unsupervised clustering models based privacy attack. Given known information about labels, the unsupervised clustering model attempts to divide a dataset into 2 clusters. In a privacy attack, it tries to form 2 groups of original datapoints and generated datapoints. If the clustering algorithm can decide with certainty the membership of a datapoint in either of the two clusters, it is a privacy risk. We use the k-means clustering algorithm to divide a mixed dataset of original datapoints and generated datapoints from our models into two groups. The accuracy of the clustering model was **0.53**. The probability of the model correctly identifying whether or not datapoint is original is close to random chance. Thus showing that our model can resist such adversarial attacks that try to identify original datapoints in a shared dataset.

4 RELATED WORK

In general, there is a lot of evidence of GANs being used for synthetic data generation and translation in image and text data [4, 23, 26]. However the properties of system or device data makes it distinct from image and text data. The system data is usually tabular i.e. contains a mix of continuous and discrete variables and in some cases the sequence of consecutive rows in the data is important. A conditional generator model can address the issue of mixed attributes in tabular data by seeking to minimize the distance between generated and real data given a fixed value of the discrete variable [3, 12, 24, 25]. System data, specifically network activity data, generated using conditional GANs have been shown to replace actual datasets in downstream tasks like network intrusion detection [6, 12, 20]. However, conditional GANs still rely on values of discrete attributes previously seen in the training set. This limits the capability of producing novel synthetic sets over a wider range of values. Additionally GANs can not maintain sequencing in

generated data. In this paper, we use knowledge infusion to address these limitations of GANs.

Knowledge graphs are ideal for storing contextual information about distributed systems that can enhance learning. The SOUPA ontology [1] captures information to support pervasive computing applications. The DAMLJessKB [11] captures attack vectors in a target centric ontology. The Unified Cybersecurity Ontology (UCO) [21] is a unified ontology for cyber situational awareness in cybersecurity systems that has been shown to enhance contextual awareness in ML systems [16, 19]. Hui et al. [10] proposed a knowledge enhanced GAN to generate IoT traffic data for devices from multiple manufacturers using KG that captures manufacturer and contextual information for IoT devices. In this work, we are looking to replace network traffic data as observed by manual testers using knowledge infused learning. Manual testers typically do not look at raw network data but process them through network traffic capture tools like Wireshark. Hence, our KG needs to maintain the semantics of data as presented by such network capture tools. Hence, we create a novel KG that captures network traffic events as interpreted by network capture tools.

5 CONCLUSION

Cybersecurity threats are dynamic and constantly evolving. Continuous monitoring of system data plays a crucial role in building secure IT infrastructure by safeguarding against emerging threats, identifying vulnerabilities, and detecting potential points of failure. It requires the of advanced tools to continually monitor and analyse networks, systems, and configurations. However, organizations encounter difficulties when adopting modern techniques like Machine Learning (ML) due to the associated privacy and security risks linked to sharing internal data. Privacy-preserving data generation involves the creation of new data that maintains privacy while still being useful as an alternate for sensitive data. Generative models, specifically Generative Adversarial Networks (GANs), have emerged as potential solutions for privacy-preserving synthetic data generation. However, standard GANs exhibit limitations in generating realistic system data. To address these limitations, we propose a novel model that uses Knowledge Infusion to overcome common limitations of Privacy Preserving Data Generation. By utilizing domain-specific knowledge, such as valid attribute ranges and constraints, the proposed model enhances the generation of synthetic data while ensuring privacy preservation. We show the validity of our framework against Network activity dataset. In future, we want to extend this work to other system data and datasets from other domains.

REFERENCES

[1] 2004. Soupa: Standard ontology for ubiquitous and pervasive applications. In *The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004.* IEEE, 258–267.

[2] Mamoun Alazab, Swarna Priya RM, Parimala M, Praveen Kumar Reddy Maddikunta, Thippa Reddy Gadekallu, and Quoc-Viet Pham. 2022. Federated Learning for Cybersecurity: Concepts, Challenges, and Future Directions. *IEEE Transactions on Industrial Informatics* 18, 5 (2022), 3501–3509. https://doi.org/10.1109/TII.2021.3119038

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*. PMLR, 214–223.

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. 2018. Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096* (2018).

[5] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM computing surveys (CSUR)* 41, 3 (2009), 1–58.

[6] Adriel Cheng. 2019. PAC-GAN: Packet Generation of Network Traffic using Generative Adversarial Networks. In *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. 0728–0734. https://doi.org/10.1109/IEMCON.2019.8936224

[7] Kelton AP Da Costa, João P Papa, Celso O Lisboa, Roberto Munoz, and Victor Hugo C de Albuquerque. 2019. Internet of Things: A survey on machine learning-based intrusion detection approaches. *Computer Networks* 151 (2019), 147–157.

[8] Soham Dasgupta, Aritran Piplai, Anantaa Kotal, and Anupam Joshi. 2020. A comparative study of deep learning based named entity recognition algorithms for cybersecurity. In *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2596–2604.

[9] Derui Ding, Qing-Long Han, Yang Xiang, Xiaohua Ge, and Xian-Ming Zhang. 2018. A survey on security control and attack detection for industrial cyber-physical systems. *Neurocomputing* 275 (2018), 1674–1683.

[10] Shuodi Hui, Huandong Wang, Zhenhua Wang, Xinghao Yang, Zhongjin Liu, Depeng Jin, and Yong Li. 2022. Knowledge Enhanced GAN for IoT Traffic Generation. In *Proceedings of the ACM Web Conference 2022*. 3336–3346.

[11] Anupam Joshi, Tim Finin, John Pinkston, et al. 2003. A target-centric ontology for intrusion detection. In *Workshop on Ontologies in Distributed Systems, held at The 18th International Joint Conference on Artificial Intelligence*.

[12] Anantaa Kotal, Aritran Piplai, Sai Sree Laya Chukkapalli, and Anupam Joshi. 2022. PriveTAB: Secure and Privacy-Preserving sharing of Tabular Data. In *Proceedings of the 2022 ACM on International Workshop on Security and Privacy Analytics*. 35–45.

[13] Wenke Lee and Salvatore J Stolfo. 2000. A framework for constructing features and models for intrusion detection systems. *ACM transactions on Information and system security (TiSSEC)* 3, 4 (2000), 227–261.

[14] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).

[15] Sumit More, M Lisa Mathews, Anupam Joshi, Tim Finin, et al. 2012. A semantic approach to situational awareness for intrusion detection. In *Proceedings of the National Symposium on Moving Target Research*.

[16] Sandeep Nair Narayanan, Ashwinkumar Ganesan, Karuna Joshi, Tim Oates, Anupam Joshi, and Tim Finin. 2018. Early detection of cybersecurity threats using collaborative cognition. In *2018 IEEE 4th international conference on collaboration and internet computing (CIC)*. IEEE, 354–363.

[17] Aritran Piplai, Mike Anoruo, Kayode Fasaye, Anupam Joshi, Tim Finin, Ahmad Ridley, et al. 2022. Knowledge guided Two-player Reinforcement Learning for Cyber Attacks and Defenses. In *International Conference on Machine Learning and Applications*.

[18] Aritran Piplai, Anupam Joshi, and Tim Finin. 2023. Offline RL+ CKG: A hybrid AI model for cybersecurity tasks. In *Proceedings of the AAAI 2023 Spring Symposium on Challenges Requiring the Combination of Machine Learning and Knowledge Engineering (AAAI-MAKE 2023)*.

[19] Aritran Piplai, Sudip Mittal, Anupam Joshi, Tim Finin, James Holt, and Richard Zak. 2020. Creating cybersecurity knowledge graphs from malware after action reports. *IEEE Access* 8 (2020), 211691–211703.

[20] Mustafizur R Shahid, Gregory Blanc, Houda Jmila, Zonghua Zhang, and Hervé Debar. 2020. Generative deep learning for Internet of Things network traffic generation. In *2020 IEEE 25th Pacific Rim International Symposium on Dependable Computing (PRDC)*. IEEE, 70–79.

[21] Zareen Syed, Ankur Padia, Tim Finin, Lisa Mathews, and Anupam Joshi. 2016. UCO: A unified cybersecurity ontology. *UMBC Student Collection* (2016).

[22] Daniele Ucci, Leonardo Aniello, and Roberto Baldoni. 2019. Survey of machine learning techniques for malware analysis. *Computers & Security* 81 (2019), 123–147.

[23] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 8798–8807.

[24] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2019. Modeling tabular data using conditional gan. *Advances in Neural Information Processing Systems* 32 (2019).

[25] Lei Xu and Kalyan Veeramachaneni. 2018. Synthesizing tabular data using generative adversarial networks. *arXiv preprint arXiv:1811.11264* (2018).

[26] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. 2017. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*. 5907–5915.