Bridging Privacy and Utility: An In-Depth Survey of Synthetic Data Generation Methods

Dr.Nagabhushan SV, Associate Professor, Dept. of CSE, BMSIT&M

***Abstract*—Generating synthetic data has become an important technique in data science that provides solutions to many challenges such as private data, rare data, and rich information. This research explores the diversity of computing techniques, from artificial intelligence techniques such as artificial neural networks (GANs), generalized instruction tuning and variable auto-encoders (VAEs) to legal rendering, live cloning and data protection technology. An overview of each method is provided and its content, advantages, limitations, and practical applications in various fields are discussed. Through comparative analysis, this article evaluates the advantages and disadvantages of each method and provides insight into their suitability for various applications. It also discusses the challenges and future directions in the development of synthetic materials and provides recommendations to researchers and professionals. This research is important for understanding the state of the art in synthetic materials design and informs future research in this rapidly changing field.**

***Keywords*—synthetic data, AI techniques, comparative analysis**

I. Introduction

In the age of big data and advanced analytics, access to good data is essential to foster innovation and decision-making in many fields. However, concerns about data privacy, rarity, and diversity often hinder the availability of real-world data for analysis and research. Synthetic profiling has emerged as a promising solution to these challenges, allowing researchers and professionals to generate accurate information and privacy for a variety of applications.

Data subjects often de-identify or anonymize data in various ways, including removing personal characteristics (e.g. name and address), scrambling (e.g. at birth) in order to provide sensitive information to others or split the changes into different categories to have more people in each category [1]. Although the additional data contained in legally anonymized data will not be used for personal identification purposes, it will contain sufficient data to confirm identity when associated with other data (such as social media platforms). Efforts to determine the effectiveness of de-identification techniques have been unsuccessful, especially in the context of big data [2].

Synthetic data generation has been researched for nearly three decades [4] and has applications in many domains [5, 6], including patient data [7] and medical data (EHRs) [8, 9]. It can be a useful tool in situations where real data is expensive, scarce, or unavailable. Although obtaining new knowledge directly from synthetic materials is not possible or advisable in some applications, it can still be used for many secondary applications such as for learning.

Depending on the purpose, synthetic data can replace real data, augment real data, or be used as a surrogate for rapid investigations [3].

In the information science and intelligence business, synthetic information has become an important tool to solve special, rare and diverse information problems. This introduction provides a brief overview of synthetic dataset generation methods: Generative Adversarial Networks (GANs), Generalized Instruction Tuning, Variable Auto-encoders (VAEs), Rules Engine, Entity Cloning, and Data Masking techniques. Although the rules-based approach limits the ability to capture complex patterns, it provides flexibility by allowing users to define business rules for generating data. This is especially important when there is a relationship between the data stored. Although these methods involve a balance between confidentiality and the use of the power of data paper, they cover anonymous data or move sensitive data to confidentiality while protecting the dataset. Understanding: principles, practices, and trade-offs. Understanding these processes allows scientists and engineers to make informed decisions when designing synthetic products for multiple users.

II. GENERATIVE AI TECHNIQUES

Generative adversarial networks (GAN), generalized instruction tuning, and variable auto-encoders (VAE) are important technologies in the field of generative intelligence. Each method uses different methods to learn from existing data and create synthetic data to meet a variety of applications in data science and beyond.

A. *Generative Adversarial Networks (GANs)*

GAN consists of two neural networks: the generator and the supervisor involved in the minimax game. While the generator creates the data model, the person watches the difference between real and synthetic models. By providing feedback, GANs improve the machine's ability to generate real information to fool the discriminator. This process produces synthetic materials that resemble the original materials.

To further explain how the network is trained, the training is split into training the discriminator and the generator separately. Training the discriminator is to create a data set consisting of the events generated by G and the content of the original data. The discriminator produces a probability (a continuous value between 0 and 1) that indicates whether the observation belongs to the original data (0 means the discriminator is 100% sure that the given rate bound is synthetic, while 1 means completely different) [10].

Given the feedback from the discriminator (e.g. unemployment rate), the producer attempts to improve the discriminator. As training is done, G uses the results of D to create better models, i.e. samples like real data. As the information produced by G becomes more accurate, D is also improved so that it can be better determined whether the model is real or synthetic. Therefore the two networks can improve each other and ideally G will be able to follow the data distribution and D will be 12 each, for example D differs from the real analysis and their Random generation is to predict the result. In this ideal case, G successfully redistributed the original data by lying to D [10].

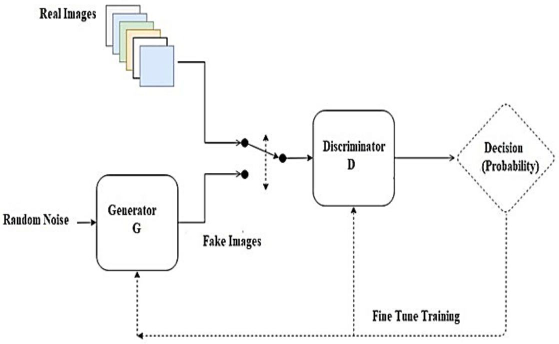

Fig. 1. A GAN Diagram

Some challenges often hinder successful learning of GANs, regardless of the details or the design adopted. When the generator and discrimination fail to reach equilibrium, a loss of emissions occurs and there is no change in unemployment during the study period. Species collision occurs when the generator focuses on a few species in the target distribution and ignores others, creating limited diversity in its output. When the discriminator is too good, data loss occurs and the minimum gradient signal is provided to drive the generator update. Vanishing gradients occur when the gradients are too small to change negative patterns, hindering the progression of learning. Hyper-parameter tuning involves optimization of various parameters such as learning rate and network architecture to achieve GAN performance. Solving these issues requires careful experimentation, new ideas, and optimization strategies to ensure GAN training is stable and effective across a variety of applications and materials.

Generative Adversarial Networks (GANs) have emerged as a powerful method for generating synthetic data across multiple disciplines. One use for GANs is the generation of realistic images, such as those featuring humans, animals, or landscapes. These synthetic images can be used for many purposes, including creating various datasets for artistic or creative endeavors, developing computer vision models, and creating lifelike visual content for video games and virtual reality. Additionally, GANs are utilized to generate study-worthy artificial medical images, which simplifies the development and testing of medical imaging techniques without the need for large, annotated datasets. GANs are also employed in data augmentation, which generates additional training examples to improve the robustness and wider applicability of machine learning models.

B. *Generalized Instruction Tuning*

Large Language Models (LLMs) provide the unprecedented ability to understand and produce human-like text. By expanding the sample size and data size , LLM can better predict the next character and perform certain activities with certain teaching methods . Intelligence does not then translate directly into human advice [11].

Natural language processing (NLP) datasets provided by instructions are used to refine LLMs before they are applied to fresh (NLP) jobs [12]. However, the restricted set of NLP tasks available limits the generalization capacity of tailored LLMs [13, 14] in real-world scenarios. Self-instruct [12] is a cheap method of creating artificial instruction tuning datasets, which generate new instructions by using randomly selected instructions from the pool to few-shot prompt an LLM (like text-davinci-002). First, a small set of seed instructions written by humans is used to start the process. Unfortunately, the diversity of generated instructions remains a challenge because few-shot prompting tends to provide new instructions that resemble past ones. Moreover, producing seed instructions of superior quality requires a substantial amount of human labor and knowledge.

Evolve-Instruct [15] improves self-instruct by using LLMs to add different rewriting processes (data argumentation) to existing instruction tuning datasets. Therefore, the scope of activities or domains that these improved datasets are capable of covering is limited by the original datasets used as input. Research has also concentrated on developing activity- or domain-specific datasets for instruction modification.

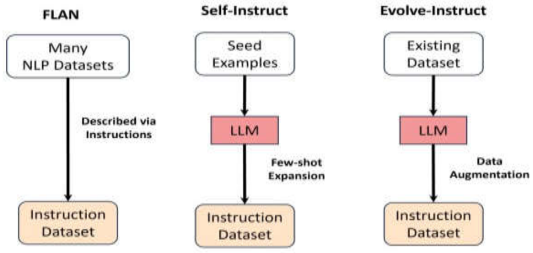

Fig. 2. Different methods of Generalized Instruction Tuning

Scalability, privacy, and communication overhead are problems for FLAN, which uses federated learning techniques, particularly in large-scale distributed systems. Depending on the chosen self-supervised learning target, self-instruct techniques can produce synthetic samples of varying quality. Additionally, while they may perform well with some forms of data, such as text or photographs, they may not do well with structured tabular data. Evolve-Instruct is based on computationally intensive evolutionary algorithms, which may have issues with constrained generalization if the initial population of synthetic data samples is not diverse. These techniques offer intriguing avenues for producing synthetic data, despite the possibility of issues; nonetheless, one must carefully consider their limitations in specific use cases and data domains.

Generalized Instruction Tuning (GIT) is a novel approach to synthetic data generation that leverages machine learning models to infer the underlying data distribution from a sparse collection of requirements or instructions. One practical application of GIT is the generation of synthetic text data for tasks related to natural language processing. Upon obtaining high-level instructions or guidelines, such as intended content structure, tone, or attitude, GIT can generate a variety of contextually relevant text data. This synthetic text can be used to improve text databases, train language models, or create realistic conversation transcripts for chatbot production. Furthermore, social media analytics, tailored recommendation systems, and content production for marketing initiatives could be beneficial for GIT. In these areas, generating customized text data is crucial for establishing a connection with customers and understanding their preferences. In addition, GIT can be used to generate synthetic time series data for forecasting models, anomaly detection, and financial simulations. This enables researchers and practitioners to experiment with different scenarios and study the behavior of systems in varied surroundings.

C. *Variable Auto-Encoders (VAE)*

A variational auto-encoder (VAE) is a type of semi-supervised/self-supervised neural network design that is a member of the Auto-encoder family. When it comes to representing a dataset, the VAEs are similar to neural generative models that use a Gaussian distribution by first encoding the input data into a reduced dimensional space (a Gaussian distribution density), then decoding a sample from this distribution back to the original input. Put another way, this type of neural network tries to reproduce the input even in the face of severe limitations (fewer nodes in hidden layers) [16].

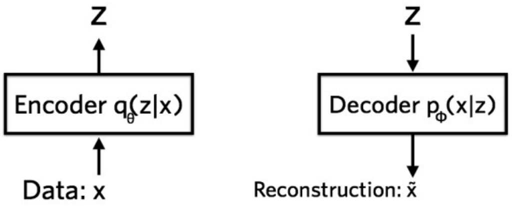

Fig. 3. Visual structure of auto-encoder

Variational auto-encoders, or VAEs, are useful tools for building synthetic datasets, but they have a number of disadvantages. One major issue is their tendency to produce fuzzy or low-quality samples, especially when compared to other generative models such as Generative Adversarial Networks (GANs). This is due to the fact that the model is often driven to generate samples that are found in the dense regions of the data distribution by the VAE objective function, which often results in outputs that are less diverse or realistic. Additionally, VAEs may have trouble accurately capturing complex data distributions, especially in high-dimensional spaces or datasets with intricate inter-variable linkages.

Another challenge is that, by default, VAEs operate at the level of the entire dataset rather than focusing on specific aspects, making it difficult to control certain features or qualities of the generated samples. Despite these limitations, VAEs are nevertheless a helpful tool for producing synthetic data, particularly when combined with other techniques or when interpretability and latent space representations are important considerations.

Variational Auto-encoders (VAEs) offer versatile applications for generating artificial intelligence across multiple domains. One use case for VAEs in the healthcare sector is the creation of artificial medical images, such as MRI or X-ray scans, to train deep learning models without compromising patient privacy. By learning the underlying distribution of real medical images, VAEs may produce synthetic images that closely resemble the original data while retaining significant features and statistical aspects. This enables researchers and clinicians to develop and evaluate image-based diagnostic algorithms, assess therapeutic efficacy, and conduct large-scale studies using a range of representative datasets. In the banking and finance sectors, VAEs are also helpful in producing artificial financial data for risk management and fraud detection. By merging transactional data while preserving its statistical characteristics, VAEs make it easier to create and test fraud detection algorithms and predictive models in a secure and compliant manner. Furthermore, manufacturing processes can benefit from the utilization of synthetic sensor data produced by VAEs to improve product quality control, optimize predictive maintenance plans, and simulate real-world operating conditions for industrial machinery and equipment.

III. RULES ENGINE

Generating synthetic data using a rules engine involves defining and applying user-defined business rules to create data that adheres to specific criteria or conditions.

- Rule Definition: Users define rules that govern the generation of synthetic data based on their domain knowledge and specific requirements. These rules can include constraints, transformations, or conditional logic that dictate how data should be generated.

- Data Schema Mapping: The rules engine maps the defined rules to the data schema or structure of the target dataset. This mapping ensures that the generated data aligns with the expected format and attributes of the original dataset.
- Data Generation: The rules engine processes the defined rules to generate synthetic data instances that satisfy the specified criteria. This may involve generating data from scratch or modifying existing data to conform to the rules.
- Quality Assurance: Generated data undergoes quality checks to ensure that it meets predefined standards and accurately represents the underlying data distribution. This may involve validating data consistency, completeness, and adherence to business rules.
- Iteration and Refinement: Users can iteratively refine and adjust the rules based on feedback and validation results to improve the quality and relevance of the synthetic data.

Overall, using a rules engine for synthetic data generation provides flexibility and control over the data generation process, allowing users to tailor the generated data to their specific needs and constraints. It's commonly used in industries such as finance, healthcare, and retail, where strict regulations, privacy concerns, and complex business logic necessitate customized data generation approaches.

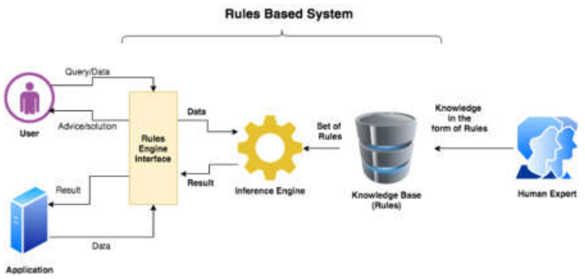

Fig. 4. Rules Engine

Rules engines provide a systematic framework for generating synthetic data based on user-defined business rules. Regretfully, their utility is limited because it requires a great deal of topic expertise and effort to design and maintain appropriate rules. Larger and more diverse datasets may provide scalability issues, making it more challenging for rules-based approaches to adapt to changing data requirements. Moreover, complicated data distributions may be difficult for rules engines to comprehend, leading to the creation of artificial datasets that are either diverse or unrepresentative of real-world scenarios. Despite these challenges, rules-based generation is nevertheless helpful for ensuring privacy and regulatory compliance in a range of businesses. In order to enhance the precision and authenticity of artificial intelligence models, rules engines can be combined with complementing methods such as data masking. Furthermore, rule sets can be updated in response to feedback in order to lessen some of the drawbacks of this approach.

Rules-based synthetic data generation approaches have practical applications across several industries and use cases. Healthcare organizations employ rules engines to generate fictional patient data while adhering to stringent privacy regulations like HIPAA. Similar to this, in finance, pre-established rules are utilized to generate synthetic financial transaction data that models different situations for compliance testing and risk assessment. Rules-based synthetic data synthesis in retail aids in inventory management and demand forecasting by producing synthetic sales and customer data. Rules engines are also utilized in the manufacturing industry to generate synthetic sensor data that improves predictive maintenance algorithms and increases output.

IV. ENTITY CLONING

Entity cloning is a method used to generate synthetic data by extracting and replicating data from a single business entity (e.g., customer, product, transaction) across various sources while maintaining referential integrity and privacy. The process involves the following steps:

- Data Extraction: Extract relevant data pertaining to the target business entity from different source systems or databases.
- Data Masking: Mask sensitive or personally identifiable information (PII) within the extracted data to ensure privacy compliance and data security.
- Cloning: Clone the extracted data of the business entity, creating multiple instances with unique identifiers while maintaining the relationships and integrity of the data.
- Anonymization: Further anonymize the cloned data to prevent re-identification of individuals while retaining the statistical properties and characteristics of the original data.
- Quality Assurance: Validate the synthetic data to ensure its accuracy, consistency, and suitability for intended use cases.

Entity cloning ensures that synthetic data accurately represents real-world scenarios while addressing privacy concerns and maintaining data integrity. This method finds applications in various domains, including finance, healthcare, retail, and manufacturing, where large-scale datasets are required for analysis, testing, and model training while ensuring compliance with data privacy regulations.

It is important to consider the many constraints associated with the entity cloning process when generating synthetic datasets. First off, entity cloning could have problems introducing variance or unpredictability to the generated dataset, despite the fact that it is an effective method for swiftly generating massive volumes of data by copying and extracting information from previous entities. Because of this lack of diversity, fake datasets may not fully capture the breadth and complexity of real-world data, which could diminish the effectiveness of later applications such as machine learning model training. Additionally, if the cloned data is not properly anonymized or disguised, privacy issues may develop as entity cloning entails simply reproducing data from existing entities without sufficient privacy safeguards. Furthermore, when dealing with complex data schemas or connected entities, entity cloning may find it challenging to maintain relational integrity across the generated dataset. Entity cloning is an efficient and simple method for producing synthetic data overall, but it has limitations with regard to privacy protection, diversity, and relational integrity. These considerations make it crucial to consider your options carefully and employ alternative tactics before utilizing this technique.

Entity cloning is a useful technique that may be applied in a variety of contexts and quickly creates artificial datasets with referential integrity maintained. Synthetic patient datasets can be produced for AI model training in the medical field without sacrificing patient privacy. Financial organizations create synthetic financial transaction data via entity cloning to detect fraud while protecting consumer confidentiality. This approach simulates equipment or inventory records to predict breakdowns or stock shortages, which helps with predictive maintenance and inventory optimization in the manufacturing sector. Organizations can easily optimize operations and train models by copying records and introducing variances. Notwithstanding its benefits, entity cloning has drawbacks, such as the difficulty to produce completely new data and certain privacy concerns if improperly concealed.

V. DRAWBACKS OF USING SYNTHETIC DATA

Despite its benefits, using synthetic data has a number of disadvantages. First off, compared to real-world data, synthetic data might not be as realistic, which could result in inaccurate modeling and analysis. Furthermore, methods for creating synthetic data may add biases or find it difficult to generalize across many contexts, which could affect how resilient the models are. Verifying synthetic data for correctness and quality is difficult and needs close examination. Concerns about privacy also surface because, even if synthetic data is meant to preserve privacy, its management could still leave users vulnerable to re-identification. Furthermore, synthetic data might not account for every edge case found in real-world data, which would reduce its usefulness in some situations. Last but not least, producing high-quality synthetic data is more resource-intensive than using real data since it requires a lot of computer power, knowledge, and time.

REFERENCES

[1]Ursin G, Sen S, Mottu J-M, Nygård M. Protecting privacy in large datasets—first we assess the risk; then we fuzzy the data. Cancer Epidemiol Prev Biomark. 2017; 26(8):1219–24.

[2]El Emam K, Jonker E, Arbuckle L, Malin B. A systematic review of re-identification attacks on health data. PLoS ONE. 2011; 6(12):1–12.https://doi.org/10.1371/journal.pone.0028071

[3]Goncalves, A., Ray, P., Soper, B. *et al.* Generation and evaluation of synthetic patient data. *BMC Med Res Methodol* **20**, 108 (2020). https://doi.org/10.1186/s12874-020-00977-1

[4]Rubin D. B.Discussion: Statistical disclosure limitation. J Off Stat. 1993; 9(2):461–8

[5]Drechsler J.Synthetic Datasets for Statistical Disclosure Control: Theory and Implementation. Lecture notes in statistics, vol. 201. New York: Springer; 2011.

[6]Howe B, Stoyanovich J, Ping H, Herman B, Gee M. Synthetic Data for Social Good. In: Bloomberg Data for Good Exchange Conference: 2017. p. 1–8

[7]Kim J, Glide-Hurst C, Doemer A, Wen N, Movsas B, Chetty IJ. Implementation of a novel algorithm for generating synthetic ct images from magnetic resonance imaging data sets for prostate cancer radiation therapy. Int J Radiat Oncol Biol Phys. 2015; 91(1):39–47. https://doi.org/10.1016/j.ijrobp.2014.09.015

[8]Walonoski J, Kramer M, Nichols J, Quina A, Moesel C, Hall D, Duffett C, Dube K, Gallagher T, McLachlan S. Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. J Am Med Inform Assoc. 2018; 25(3):230–8.

[9]Dube K, Gallagher T. Approach and Method for Generating Realistic Synthetic Electronic Healthcare Records for Secondary Use. In: International Symposium on Foundations of Health Information Engineering and Systems. Springer: 2014. https://doi.org/10.1007/978-3-642-53956-5_6

[10]Figueira, A.; Vaz, B. Survey on Synthetic Data Generation, Evaluation Methods and GANs. *Mathematics* **2022**, *10*, 2733. https://doi.org/10.3390/math10152733

[11]Haoran Li , Qingxiu Dong , Zhengyang Tang , Chaojun Wang , Xingxing Zhang , Haoyang Huang, Shaohan Huang, Xiaolong Huang, Zeqiang Huang, Dongdong Zhang, Yuxian Gu, Xin Cheng Xun Wang, Si-Qing Chen, Li Dong, Wei Lu, Zhifang Sui, Benyou Wang, Wai Lam, Furu Wei. Synthetic Data (Almost) from Scratch: Generalized Instruction Tuning for Language Models.

https://doi.org/10.48550/arXiv.2402.13064

[12]Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language model with self generated instructions. arXiv preprint arXiv:2212.10560, 2022.

[13]Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. Advances in Neural Information Processing Systems, 35:24824–24837, 2022.

[14]Tianwen Wei, Liang Zhao, Lichang Zhang, Bo Zhu, Lijie Wang, Haihua Yang, Biye Li, Cheng Cheng, Weiwei Lü, Rui Hu, Chenxia Li, Liu Yang, Xilin Luo, Xuejie Wu, Lunan Liu, Wenjun Cheng, Peng Cheng, Jianhao Zhang, Xiaoyu Zhang, Lei Lin, Xiaokun Wang, Yutuan Ma, Chuanhai Dong, Yanqi Sun, Yifu Chen, Yongyi Peng, Xiaojuan Liang, Shuicheng Yan, Han Fang, and Yahui Zhou. Skywork: A more open bilingual foundation model, 2023.

[15]Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. arXiv preprint arXiv:2304.12244, 2023.

[16[Ally Salim Jr. Synthetic Patient Generation: A Deep Learning Approach Using Variational Autoencoders.

https://doi.org/10.48550/arXiv.1808.06444