ISSN: 2998-8713

Review Article
Journal of data Analytics and Engineering Decision Making

Exploring the Landscape for Generative Sequence Models for Specialized Data Synthesis

Mohammad Zbeeb*, Mohammad Ghorayeb and Mariam Salman

Department of Electrical and Computer Engineering, American University of Beirut, Lebanon

***Corresponding Author**
Mohammad Zbeeb, Department of Electrical and Computer Engineering, American University of Beirut, Lebanon.

Submitted: 2024, Nov 08; **Accepted:** 2024, Dec 07; **Published:** 2024, Dec 17

Citation: Zbeeb, M., Ghorayeb, M., Salman, M. (2024). Exploring the Landscape for Generative Sequence Models for Specialized Data Synthesis. *J Data Analytic Eng Decision Making, 1*(2), 01-14.

Abstract
Artificial Intelligence (AI) research often aims to develop models that can generalize reliably across complex datasets, yet this remains challenging in fields where data is scarce, intricate, or inaccessible. This paper introduces a novel approach that leverages three generative models of varying complexity to synthesize one of the most demanding structured datasets: Malicious Network Traffic. Our approach uniquely transforms numerical data into text, re-framing data generation as a language-modeling task, which not only enhances data regularization but also significantly improves generalization and the quality of the synthetic data. Extensive statistical analyses demonstrate that our method surpasses state-of-the-art generative models in producing high-fidelity synthetic data. Additionally, we conduct a comprehensive study on synthetic data applications, effectiveness, and evaluation strategies, offering valuable insights into its role across various domains. Our code and pre-trained models are openly accessible at Github, enabling further exploration and application of our methodology.

Keywords: Data Synthesis, Machine Learning, Traffic Generation, Privacy Preserving Data, Generative Models

1. Introduction

Machine learning algorithms depend heavily on the availability and quality of training data. However, acquiring real-world data poses challenges due to privacy concerns, limited accessibility, and potential biases [1]. Consequently, synthetic data generation has attracted increasing interest, aiming to create diverse and representative datasets that mitigate issues of data scarcity, bias, and privacy [2]. In recent years, Generative Adversarial Networks (GANs) have emerged as a powerful technique for producing realistic synthetic data [3]. GANs are widely applied in fields such as image generation, network traffic modeling, and healthcare data synthesis [4]. These models replicate the statistical properties of real-world data, providing a valuable tool for augmenting datasets in cases where data is limited or sensitive [5]. Despite their impressive results, GANs face challenges. Computational complexity and training instability have been widely documented, complicating replication across domains [6]. Moreover, GANs' primary focus on unstructured data raises questions about their suitability for structured numerical data, which is often critical in fields such as cybersecurity, finance, and healthcare [5]. This has fueled demand for alternative generative models capable of efficiently handling structured data while preserving the original data's key statistical properties. Beyond GANs, Variational Autoencoders (VAEs) and other generative models have shown promise for synthetic data generation. VAEs effectively capture complex data distributions in recommendation systems and collaborative filtering [7]. However, they may lack representational power compared to GANs, especially with complex datasets [4]. Alongside methodological advancements, several studies have integrated privacy-preserving mechanisms into generative models. Differentially private GANs, for instance, generate synthetic data that maintains privacy and minimizes the risk of sensitive information leakage [2]. Such approaches are essential in sensitive domains like healthcare, where data privacy is paramount, requiring a careful balance between data quality and ethical considerations.

Existing synthetic data generation methods often focus on unstructured data or encounter challenges in specialized fields like cybersecurity and financial risk modeling. Furthermore, adversarial training and continuous distribution modeling can complicate the generation process, particularly for structured numerical data with irregularities or outliers [5]. This paper builds on existing research by exploring the potential of sequence models for synthetic data generation. Sequence models, widely used in natural language processing, present a novel approach to generating structured data by framing it as a language-modeling problem [8]. By leveraging sequence models' strengths in handling both discrete and continuous data, we aim to address limitations posed by traditional generative models, particularly in domains requiring structured, high-dimensional data. Informed by key findings from the literature, we aim to contribute to the discourse on synthetic data generation by investigating how sequence models can provide a computationally efficient alternative to established techniques. While GANs and VAEs have dominated the field, we propose that sequence models offer a flexible and scalable approach for generating high-quality synthetic data, particularly in scenarios with structured data and categorical variables [9].

2. Techniques

Here we describe our techniques for data generation and training the generators as language- based classifiers. First, we provide background on structured datasets and the data used in our experiments.

2.1 Dataset Overview

An overview of the data used in our experiments is presented in Table 1.

#	Attribute	Type	Example
1	Date First Seen	Timestamp	2018-03-13 12:32:30.383
2	Duration	Continuous	0.212
3	Transport Protocol	Categorical	TCP
4	Source IP Address	Categorical	192.168.100.5
5	Source Port	Categorical	52128
6	Destination IP Address	Categorical	8.8.8.8
7	Destination Port	Categorical	80
8	Bytes	Numeric	2391
9	Packets	Numeric	12
10	TCP Flags	Binary/Categorical	.A..S.

Table 1: Overview of Typical Attributes in Flow-Based Data [5]

The dataset used in this project contains attributes typical of unidirectional NetFlow data [5]. NetFlow data is highly heterogeneous, containing continuous, numeric, categorical, and binary attributes. Most attributes, such as IP addresses and ports, are categorical. Additionally, there is a timestamp attribute (Date First Seen), a continuous attribute (Duration), and numeric attributes such as Bytes and Packets. A key aspect of the dataset is the inclusion of TCP flags, defined here as binary/categorical. These flags can be interpreted either as six binary attributes (e.g., isSYN flag, isACK flag) or as a single categorical value, allowing flexibility in data processing and modeling across generative approaches. This diverse mix of attribute types poses challenges for synthetic data generation, especially when aiming to preserve the statistical properties of the original data while maintaining categorical, continuous, and binary relationships.

2.2 Data Transformation via CICFlowmeter - V4.0 (ISCX-FlowMeter)

For our experiments, we converted raw network traffic data into CSV format using CICFlowmeter- V4.0, formerly known as ISCXFlowMeter. CICFlowmeter is a bi-directional flow generator and analyzer for Ethernet traffic, specifically designed for anomaly detection in cybersecurity datasets [5]. CICFlowmeter has been extensively used in well-known cybersecurity datasets, including:

• Android Adware-General Malware dataset (CICAAGM2017),
• IPS/IDS dataset (CICIDS2017),
• Android Malware dataset (CICAndMal2017), and
• Distributed Denial of Service (CICDDoS2019).

By leveraging CICFlowmeter, we extracted 80 features from each flow, compiling a compre- hensive set of flow-based features in CSV format. This structured tabular data includes attributes such as source and destination IP addresses, transport protocols, port numbers, byte and packet counts, TCP flags, and other network metrics.

The conversion of network flows into structured tabular data is crucial for our approach, as it allows for systematic analysis and modeling. The resulting dataset, with its rich set of 80 features, provides the necessary structured format for advanced machine learning techniques, enabling effective synthetic data generation while preserving relationships between features. Structured data in this format facilitates the use of sequence models and other generative techniques that rely on well-organized, tabular data representations [10].

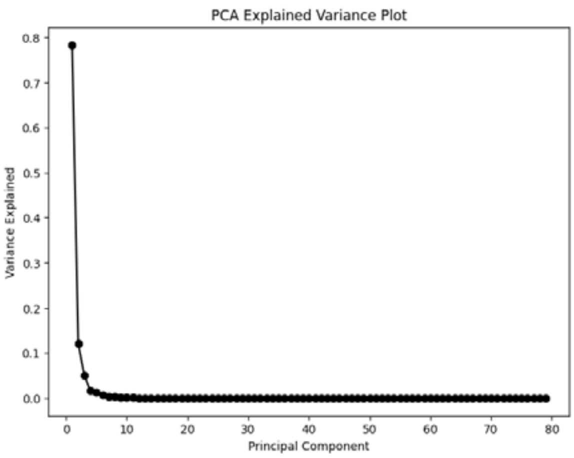


Figure 1: PCA Explained Variance Plot: The majority of the variance is captured by the first few principal components, indicating that much of the data's complexity can be explained by a small subset of components.

2.3 Data Transformation to Text Domain - Symbolic Encoding

Our preliminary exploratory data analysis revealed significant complexity within the dataset, as evidenced by high variance in certain features and the considerable number of unique values across columns. This complexity makes the dataset unsuitable for traditional statistical sampling or simple data synthesis techniques, supporting the exploration of advanced synthetic data generation models. To enhance the representational quality and address the challenges posed by this complexity, we applied a novel encoding strategy, transforming the dataset from a numeric to a symbolic, textual domain. Specifically, each numeric feature was discretized into intervals, with each interval represented by one of 49 unique symbols. Each symbol corresponds to a 1% range of the respective feature's values, resulting in a robust dataset of 30,000 examples. Each example can be considered analogous to a sentence in the symbolic domain [9]. This transformation repositions the data generation task as a classification problem rather than continuous regression. By encoding each data point as a sequence of symbols, we frame the task as the prediction of the next symbol in a sequence, given a preceding set of symbols, analogous to language modeling tasks in NLP [8]. The dataset, now framed in a discrete symbolic space, facilitates the use of classification algorithms designed for categorical outputs, aligning well with sequence models.

Algorithm 1: Symbolic Encoding Strategy for Dataset Transformation

Input: Numerical dataset $\mathcal{D}$ with N elements
Output: Transformed dataset $\mathcal{D}'$ consisting of 30,000 examples, each represented as a sequence of symbols
Initialize: Define a set $\mathcal{S}$ of 49 distinct symbols, each representing a 1% interval of the data range;
Divide: Partition the range of each numerical feature into 49 equal intervals corresponding to the symbols in $\mathcal{S}$;
foreach *element* $e \in \mathcal{D}$ **do**
  Determine the interval to which e belongs;
  Map e to the corresponding symbol $s \in \mathcal{S}$;
end
Augment: For each sequence of data points, prepend a designated start symbol to indicate the beginning of a packet;
return $\mathcal{D}'$: Transformed dataset as sequences of symbols;

3. Problem Framing

Our study frames the data generation task as the prediction of the next symbol in a sequence, given the current token. Let x represent the current token, and y the next symbol to be predicted. The probability mass function (PMF) for the random variable y, conditioned on x, is given by

P, where $P\ y\ x$ represents the probability of the next symbol y, given the current token x. Our goal is to maximize $P\ y = y_{\text{true}}$, where y_{true} is the true label of the next token.

We frame this task as a classification problem, not a regression problem. Although one might bypass text transformation by regressing the output directly, regression introduces challenges, especially when managing high-dimensional, continuous outputs with complex data structures [11].

Classification, by contrast, allows the model to discretize decision-making and capture the data's discrete nature effectively. In cases where classes occupy distinct manifolds within the data space, classification models can partition the space, yielding probabilistic predictions and clearer boundaries.

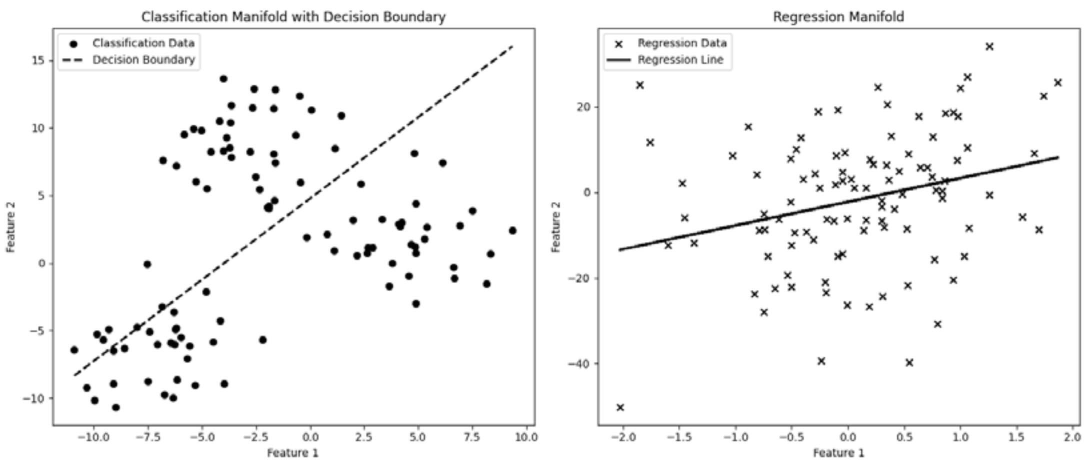


Figure 2: Comparison of Classification and Regression Manifolds. The left plot represents the classification problem with a decision boundary, while the right plot shows the regression problem with a fitted regression line.

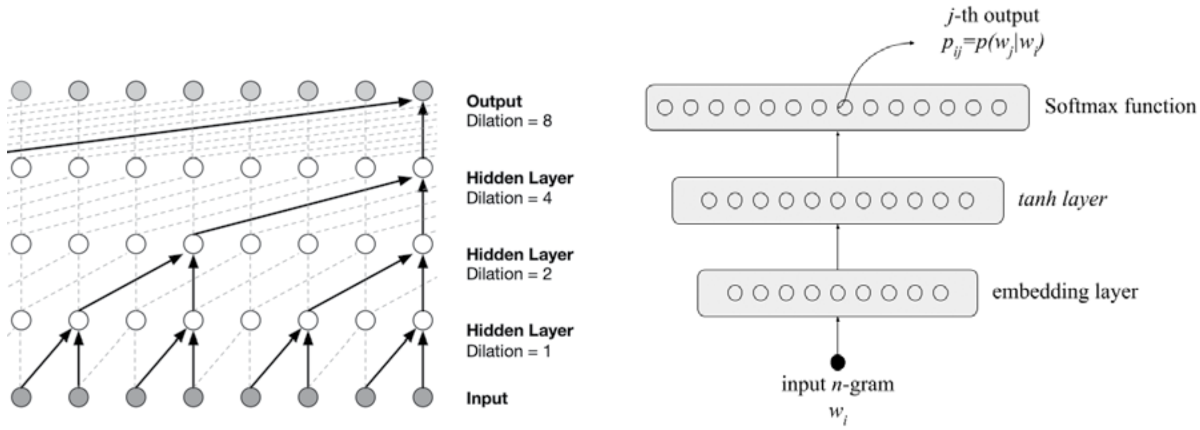
4. Overview of Sequence Models Employed in Our Study

4.1 WaveNet-Enhanced Neural Probabilistic Language Model

We employed the WaveNet architecture to enhance a neural probabilistic language model, leveraging its capability to capture intricate sequential dependencies within data. This integration advances language modeling for synthetic data generation. Neural probabilistic language models, initially introduced by Bengio et al. [8]. learn distributed token representations and predict sequences based on contextual probabilities. By integrating the WaveNet architecture, developed by Google, we extend this foundational approach [12].

WaveNet's use of **causal convolutions** ensures temporal consistency in predictions—essential for modeling sequential data tasks. The architecture predicts each token based on preceding context, enabling effective capture of linguistic structure and nuances.

$$y_t = f(x_{t-k}, x_{t-k+1}, \ldots, x_t) = \sum_{i=0}^{k} w_i \cdot x_{t-i}, \quad \text{for } t \geq k$$



(a) WaveNet Architecture [van den Oord et al., 2016]

(b) Neural Probabilistic Language Model with Causal Convolutions

Figure 3: Architectures Used in Our Study

4.2 Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) effectively process sequential data by maintaining a "memory" of previous inputs, achieved through feedback loops in the architecture. This enables RNNs to learn sequence patterns and relationships, producing coherent and contextually relevant text. Our RNN architecture leverages these capabilities, processing encoded data sequences and capturing dependencies within each 10-character segment.

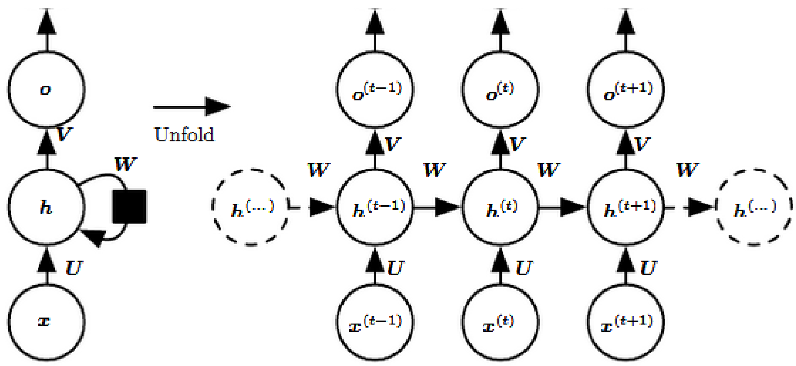


Figure 4: A Recurrent Neural Network

4.3 An Attention-Based Decoder - Transformer

The Transformer sets itself apart from traditional neural networks by avoiding recurrent mechanisms and instead leveraging self-attention, which weighs the importance of different tokens in an input sequence in parallel. This enables efficient parallel processing and better handling of long-range dependencies [9].

Our Transformer architecture employs an embedding layer with size 64 per symbol, followed by 4 Transformer blocks, each with 4 attention heads, capturing patterns in sequential data. Each Transformer block includes multi-head attention, feed-forward networks, and layer normalization, supporting robust learning of input sequences.

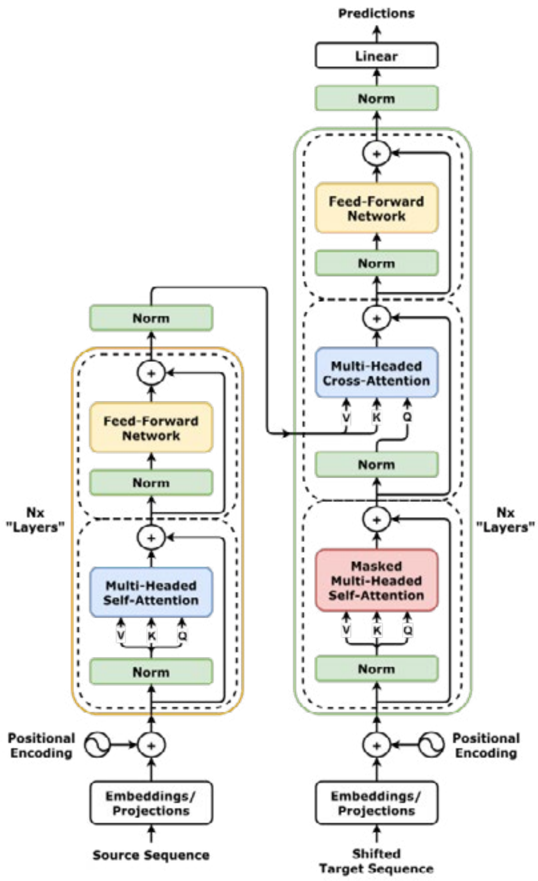


Figure 5: Transformer Architecture

5. Experiment Setup - Framework for Generating Synthetic Data

This section outlines the models used to create a novel framework for generating synthetic data. We detail the rationale behind selecting these models, discuss the appropriate loss functions, and highlight best practices in training for optimal performance. Additionally, we examine trade-offs involved in generating synthetic data, focusing on aspects of realism, diversity, and privacy preservation.

5.1 Building Intuition

The proposed framework is based on the concept of N-gram models [13]. It involves sampling from a distribution where each character is characterized by a conditional probability over the previous 1 characters.

Mathematically, this is represented as:

$$P(c_i \mid c_{i-(n-1)}, \ldots, c_{i-1}) \qquad (1)$$

This approach has limitations, such as failing to capture long-range dependencies and contextual semantics. Additionally, as n increases, the number of possible N-grams grows exponentially, leading to data sparsity and many zero-count N-grams if the training data is insufficient.

Our approach builds upon Bengio's work on Neural Probabilistic Language Models, where he proposed a neural network architecture to learn the probability distribution of word sequences [8]. By integrating these ideas with the WaveNet architecture, known for its strong performance in modeling long-range dependencies in sequential data, we aim to develop a powerful language model capable of generating highly realistic and diversified synthetic text data [12].

To clarify our methodological choices, we provide intuition behind adopting Bengio's neural network approach and emphasize its advantages in our context.

Bengio's neural network represents each word with a sampled vector and feeds it into a neural network that predicts the next word in the sequence. This prediction is achieved by turning the logits into a distribution via the softmax function, allowing sampling from this distribution. The network learns both the network parameters and the sampled distribution.

Building on this, we introduce WaveNet. Bengio's approach squashes the input through the network, making it difficult to learn long-term dependencies and positional information. In contrast, WaveNet heavily relies on dilated causal convolutions, specifically designed to capture long-term dependencies by applying multiple large dilated convolutions in parallel.

Mathematically, a dilated convolution operation for a sequence x with filter f is defined as:

$$y(t) = \sum_{k=0}^{K-1} f(k) \cdot x(t - r \cdot k)$$

Where:
• $y\ (t)$ is the output at time step t,
• K is the filter size,
• $f\ (k)$ represents the filter weights,
• $x\ (t - r \cdot k)$ are the input values with dilation rate r.

Using dilated convolutions, WaveNet can efficiently model dependencies over much longer sequences. This is achieved by applying multiple layers of dilated convolutions in parallel, with exponentially increasing dilation rates [12]. This allows the network to capture a broader context at each layer, effectively modeling long-term dependencies. Wave Net not only models bigrams but also higher-order n-grams (e.g., fourgrams) by progressively squashing the input through these convolutional layers. This gradual reduction in dimensionality captures more semantics, resulting in more realistic and contextually aware synthetic data generation [12].

Moving to other language models for synthetic data generation, we include the well-known Recurrent Neural Network (RNN). In an RNN, we hold a state h_t and pass it to the forward neuron to maintain contextual information. Mathematically, this is represented as:

$$h_t = \sigma\ (W_h h_{t-1} + W_x x_t + b)$$

where:
- h_t is the hidden state at time step t,
- W_h and W_x are weight matrices,
- x_t is the input at time step t,
- b is the bias,
- σ is the activation function (e.g., tanh or ReLU).

Next, we examine the Transformer model. In the Transformer, we maintain key, query, and value vectors. The self-attention mechanism in Transformers can be represented as [9]:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where:
- Q is the query matrix,
- K is the key matrix,
- V is the value matrix,
- dk is the dimension of the key vectors.

To enhance vector representation, we add the original vector to the value vector. Once character representation is well-learned, we can stack a probabilistic model on top of the Transformer. Even simpler models can effectively predict the next character in the sequence. For example, we can use a simple probabilistic model such as a softmax layer:

$$P(c_i \mid c_{<i}) = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$$

where $P(c_i \mid c_{<i})$ represents the probability of character c_i given the previous characters $c_{<i}$, and z_i is the logit for character c_i.

5.2 Loss

For generative tasks, where we predict the next character in a sequence from a distribution, cross-entropy loss is commonly used. This loss measures the difference between the true distribution and the predicted distribution for each packet (sequence) in the dataset.

To compute the loss over an entire sequence of packets, we sum the cross-entropy loss over all characters (or time steps) within each sequence, and then average the loss over all sequences in the dataset. The cross-entropy loss for the dataset can be defined as:

$$\mathcal{L}_{\text{cross-entropy}} = -\frac{1}{M}\sum_{j=1}^{M}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,j}\log(\hat{y}_{i,j})$$

where:

• M is the total number of sequences (or packets) in the dataset,
• C is the number of classes (characters or possible outputs for each token),
• N is the number of time steps in a sequence,
• $y_{i,j}$ is the true probability of class i in sequence j (typically 0 or 1),
• $y\hat{}i, j$ is the predicted probability of class i in sequence.

This loss formulation ensures that the model sums the loss over all time steps in each sequence, then sums over all sequences in the dataset, and finally averages the loss by the number of sequences M.

5.3 Training Practices

Generative models require additional care to ensure they produce high-quality and realistic synthetic data. Our framework includes best practices to address these needs effectively.

To address tanh issues in Bengio's approach, we reference [11]. During the forward pass, the activations passing through the tanh layer tend to be extreme, often lying on the tails at either positive one or negative one. During the backward pass, when neurons with tanh activation function update their weights, they often encounter a zero gradient. Consequently, in the update step:

$$\frac{\partial L}{\partial a} \leftarrow \frac{\partial L}{\partial a} + \frac{\partial L}{\partial y_j}U_j,$$

the neuron behaves in a shut-off mode due to a zero gradient, resulting in no weight change. To address this, we manage the standard distribution of activations entering the tanh activated layer to have a gain of $\frac{5}{3}$ over $\sqrt{\text{fan in}}$, allowing the neurons to learn normally.

For covariant shift resulting from high-dimensional datasets (curse of dimensionality), we apply batch normalization [10] as best practice for normalizing the flow (backward and forward) [14]. For a layer with d-dimensional input $\mathbf{x} = (x^{(1)}, \ldots, x^{(d)})$, we normalize each dimension $x^{(k)}$ as follows:

$$x_{\text{norm}}^{(k)} = \frac{x^{(k)} - \mathbb{E}[x^{(k)}]}{\sqrt{\text{Var}[x^{(k)}]}}$$

where:
• $x^{(k)}$ is the k-th dimension of the input,

• $E[x^{(k)}]$ is the expected value (mean) of $x^{(k)}$,
• $\mathrm{Var}[x^{(k)}]$ is the variance of $x^{(k)}$.

By normalizing each dimension of the input, batch normalization mitigates covariant shift effects, improving network stability and performance during training.

To address high initial loss in classification generative tasks, we scale output weights during initialization by a small value ( ), allowing similar probabilities across alphabets during the first pass.

Let W_{out} be the weight matrix connecting the hidden layer to the output layer, and let U_{out} be the bias vector at the output layer, initialized as:

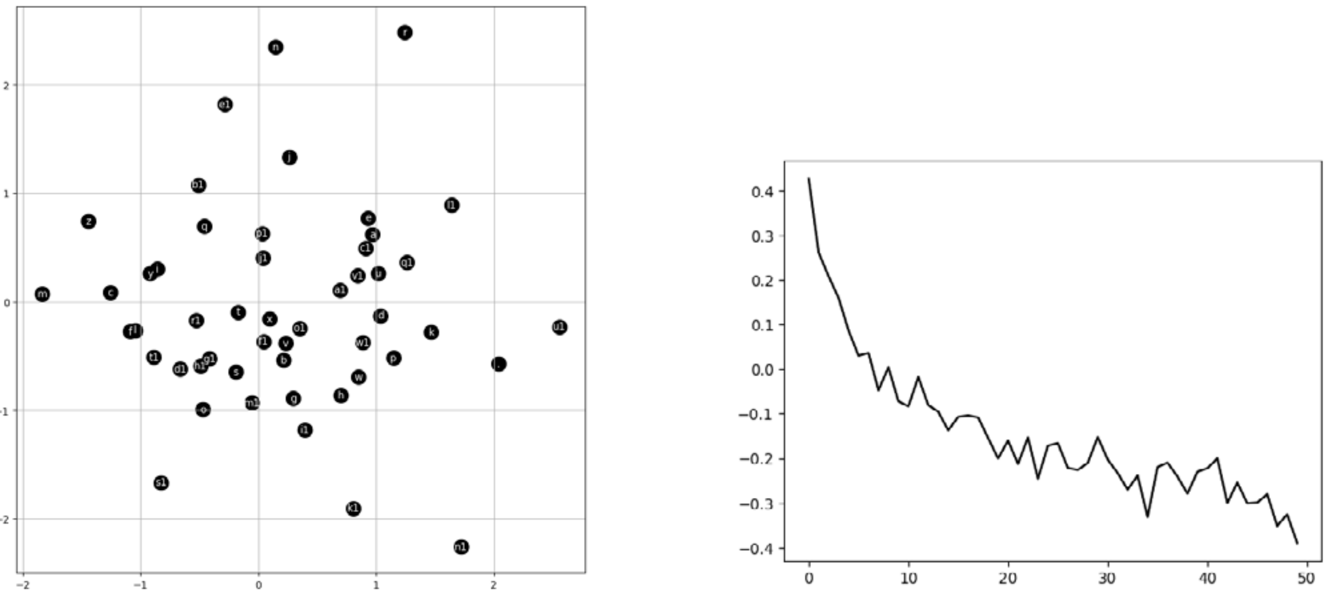
$$W_{\text{out}} \sim U(-\epsilon, \epsilon),$$

$$U_{\text{out}} \sim U(-\epsilon, \epsilon),$$

where $U(-\epsilon, \epsilon)$ is a uniform distribution. This initialization strategy ensures that weights and biases are scaled by ϵ, yielding a roughly uniform probability distribution for output activations.

$$W_{\text{out}}(i, j) \sim U(-\epsilon, \epsilon), \forall i, j,$$

$$U_{\text{out}}(i) \sim U(-\epsilon, \epsilon), \forall i.$$



(a) Latent Space

(b) General Decay of the Loss

Figure 6: Post Learning State - Symbols being learnt and Loss decay

6. Statistical Framework for Testing Generative Examples

We posit that if the distribution of generated data closely aligns with the real data distribution, it should effectively train machine learning models. To assess this, we can train a separate classifier on real data to evaluate the statistical validity of the generated data. Logically, training machine learning systems on synthetic data that closely mirrors real data should not harm performance. Overfitting typically arises when a model is overly complex relative to the dataset size, often occurring when there are more parameters than data points.

However, our generated data mitigates this risk by significantly expanding the dataset, providing a more robust foundation for model training. Mathematically, with an original dataset size of N and generated data size M, the total dataset size becomes $N + M$. Ensuring that the generated data adheres to the original distribution, $P_{\text{real}}(x) \approx P_{\text{gen}}(x)$, helps prevent model memorization of specific examples, promoting the learning of generalizable patterns [15]. Consequently, incorporating generated data enhances the model's generalization to unseen examples, rather than leading to overfitting.

Underfitting, which occurs when a classifier fails to capture underlying patterns in the data, resulting in suboptimal performance, is effectively addressed by the generated data. By training on a broader range of examples, the model can better recognize diverse features and gain a comprehensive understanding of the underlying data patterns.

Thus, we employed a one-class Support Vector Machine (SVM) with a linear kernel to determine whether the generated data is statistically similar to real data.

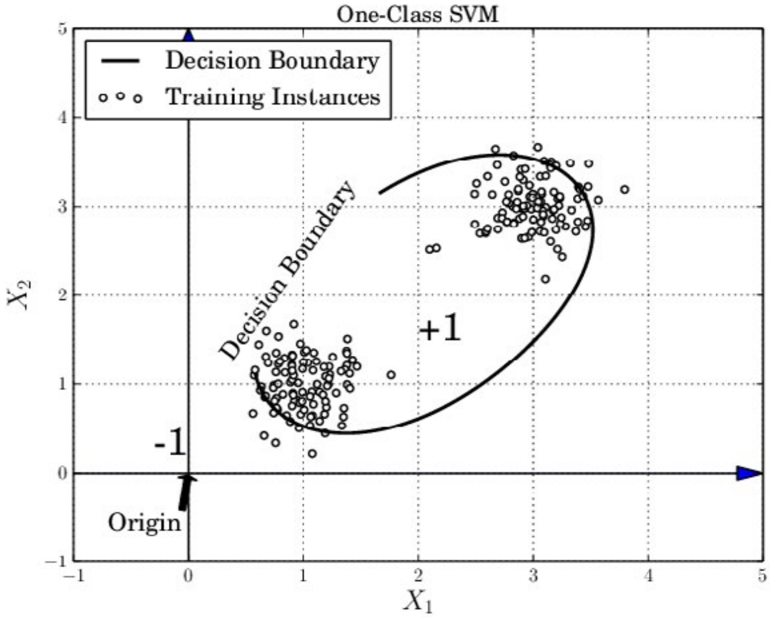


Figure 7: Pseudo Visualization of The Latent Space Post Classifying Inliers

7. Results

In our experiments, we evaluated each model's ability to generate synthetic data that closely aligns with the original data distribution. The primary evaluation metric was the percentage of inliers, defined as the proportion of generated data points that fall within the distribution of the original data.

The results indicate that while all models performed well, the Recurrent Neural Network (RNN) achieved the highest percentage of inliers at 87.9%, followed by the Transformer-based Decoder at 84.9%. WaveNet, although effective in modeling long-range dependencies, had the lowest inlier rate at 69.2%, likely due to its convolutional structure, which may not capture certain complex dependencies as efficiently as the RNN and Transformer models.

Model	Inliers (%)
WaveNet	69.2%
RNN	87.9%
Transformer Decoder	84.9%

Table 2: Inliers with Respect to Each Model

The RNN outperformed the other models in terms of generating inliers, likely due to its ability to capture sequential dependencies in the data. However, as datasets grow more complex, particularly with higher dimensionality or heterogeneity, the Transformer-based Decoder model is expected to excel. This is due to the Transformer's self-attention mechanism, which is particularly suited for handling complex dependencies and long-range interactions, which become more significant with increased data complexity. While Wave Net is designed to model long-range dependencies through dilated convolutions, it may not have been as effective for this dataset due to its convolutional architecture, which can limit its capacity to capture fine-grained patterns in structured data [12]. Nevertheless, its performance might improve with further fine-tuning and optimization.

8. Synthetic Data Generation: A Survey

Synthetic data generation has emerged as a vital solution in artificial intelligence (AI) and machine learning, offering unique advantages for both research and practical applications. In response to growing privacy concerns and limited access to real-world data, synthetic data has evolved as a powerful alternative, enabling model training, testing, and deployment without compromising sensitive information. This survey examines the diverse applications of synthetic data generation, from vision and voice technologies to business intelligence, and highlights its potential to transform data-driven fields. By synthesizing insights from recent studies, this survey aims to provide a comprehensive overview of how synthetic data is revolutionizing AI across various domains while addressing privacy and ethical considerations.

9. Applications

Synthetic data presents numerous compelling benefits, making it a highly attractive option across a wide range of applications. By streamlining the processes of training, testing, and deploying AI solutions, synthetic data enables more efficient and effective development. Furthermore, this cutting-edge technology mitigates the risk of exposing sensitive information, thereby safeguarding customer security and privacy. As researchers transition synthetic data from the laboratory to practical implementations, its real-world applications continue to expand. This section examines several notable domains where synthetic data generation substantially impacts addressing real-world challenges.

9.1 Vision

Generating synthetic data for computer vision tasks has proven highly effective, as it allows for the creation of large, diverse datasets that can be used to train models without the need for costly and time-consuming data collection efforts [16]. These synthetically generated datasets can capture a wide range of scenarios, including complex lighting conditions, occlusions, and diverse object appearances, which are crucial for developing robust vision-based systems. GANs and other generative models have emerged as powerful tools for producing such high-quality synthetic data [17-19].

In computer vision, manual labeling remains essential for certain tasks [20]. However, tasks like segmentation, depth estimation, and optical flow estimation can be particularly arduous to label manually due to their inherent complexity. To alleviate this burden, synthetic data has become a transformative tool, streamlining the labeling process significantly [21]. Sankaranarayanan et al. proposed a generative adversarial network (GAN) designed to bridge the gap between embeddings in the learned feature space, which is instrumental in Visual Domain Adaptation [22]. This methodology enables semantic segmentation across varied domains by using a generator to map features onto the image space, allowing the discriminator to operate effectively on these projections. The discriminator's output serves as the basis for adversarial losses [23]. Research has demonstrated that applying adversarial losses to the projected image space consistently outperforms applications to the feature space alone, yielding notably enhanced performance [23]. In a recent study, a team at Microsoft Research validated the efficacy of synthetic data in face-related tasks by leveraging a parametric 3D face model, enriched with a comprehensive library of hand-crafted assets [24]. This approach allowed for the rendering of training images with high levels of realism and diversity. The researchers demonstrated that machine learning models trained on synthetic data achieved accuracy comparable to models trained on real data for tasks like landmark localization and face parsing. Notably, synthetic data alone was sufficient for robust face detection in unconstrained environments [24].

9.2 Voice

The synthetic voice industry is at the cutting edge of technological progress, evolving at an unprecedented rate. The rise of machine learning and deep learning has enabled the creation of synthetic voices for applications like video production, digital assistants, and video games [25], making the process more accessible and accurate than ever. This field lies at the intersection of multiple domains, including acoustics, linguistics, and signal processing. Researchers continuously seek to enhance the accuracy and naturalness of synthetic voices. As technology continues to advance, synthetic voices are expected to become increasingly integrated into daily life, offering valuable support across various domains and enriching user experiences [25]. Earlier research involved spectral modeling techniques for statistical parametric speech synthesis, utilizing low-level, unmodified spectral envelope parameters for generating synthetic voices. These spectral envelopes are represented through graphical models with multiple hidden variables, incorporating structures like restricted Boltzmann machines and deep belief networks (DBNs) [26]. Enhancements to traditional hidden Markov model (HMM)-based speech synthesis systems have shown substantial improvements in achieving a more natural sound while reducing oversmoothing effects [27]. Synthetic data has also found applications in Text-to-Speech (TTS) systems, achieving a level of naturalness close to that of human speech [28,29]. Synthetic speech (SynthASR) has emerged as a solution for automatic speech recognition in cases where real data is sparse or limited. By integrating techniques like weighted multi-style training, data augmentation, encoder freezing, and parameter regularization, researchers have tackled issues like catastrophic forgetting. This innovative approach enables state-of-the-art training for a broad array of end-to-end (E2E) automatic speech recognition (ASR) models, reducing dependency on production data and the associated costs.

9.3 Business

The risk of compromising or exposing original data remains a constant concern, especially in the business sector, where strict restrictions govern data sharing both within and beyond the organization. This has led to an increased focus on developing financial datasets that replicate the characteristics of "real data" while safeguarding the privacy of all parties involved. Although technologies such as encryption, anonymization, and advanced privacy-preserving methods have been employed to secure original data [?], residual risks persist. Data-derived information can sometimes still be used to trace individuals, thus compromising privacy [30]. Synthetic data offers a compelling solution by removing the need to expose sensitive data, effectively ensuring privacy and security for both companies and their customers [?]. Additionally, synthetic data allows organizations faster data access by circumventing certain privacy and security protocols. Historically, institutions with large data reserves were well-positioned to assist decision- makers in tackling a range of issues. However, even internal data access was often restricted due to confidentiality concerns. Today, companies leverage synthetic data to refresh and model original datasets, generating ongoing insights that drive organizational performance improvements [?].

10. Privacy Risks and Prevention

Synthetic data generation has emerged as a key solution for data privacy and sharing in sectors where sensitive data cannot be disclosed, such as clinical, genomic, and financial domains. However, the generation of synthetic data that preserves the statistical properties of real datasets introduces privacy risks, as models may unintentionally expose underlying patterns, thereby compromising individual privacy. Membership inference attacks, for example, can identify whether specific data points were included in the training set, posing significant privacy concerns. To address these risks, privacy-enhancing methods fall into two primary categories: anonymization-based approaches and differential privacy (DP) methods.

Anonymization techniques, including k-anonymity and nearest marginal sanitization, replace sensitive information with fictitious yet realistic data, providing foundational privacy protection, though often lacking rigorous guarantees. Differential privacy methods, on the other hand, offer more robust protection by introducing noise to data, thus maintaining privacy while preserving data utility. Advanced implementations, such as GAN-based DP models (e.g., DPGAN and PATE-GAN) and local differential privacy (LDP) frameworks, support secure synthetic data generation, particularly in distributed contexts.

Alongside privacy, fairness in synthetic data is increasingly critical, as models trained on biased datasets may unfairly represent minority groups, reinforcing existing disparities. Three main approaches address fairness in synthetic data: preprocessing, which adjusts input data c to remove correlations with sensitive attributes; in-processing, which incorporates fairness constraints during model training; and post-processing, which adjusts model predictions to enhance equity. Preprocessing remains the most commonly applied fairness technique, especially for addressing subgroup imbalances through balanced synthetic datasets. Overall, privacy-enhanced synthetic data generation, coupled with fairness-aware strategies, is crucial for secure and ethical data sharing that meets both privacy and fairness standards in research and industry applications.

Paper	Privacy Technique	Model	Data Format	Notes
?	Differential Privacy	Autoencoder	Attribute	-
Lee et al. [2020]	Differential Privacy	VAE + GAN, Recurrent Autoencoder	EHR	-
Acs et al. [2018]	Differential Privacy	Generative Artificial Neural Networks	Image and Text	Kernel k-means
Jordon et al. [2018]	Differential Privacy (PATE)	GAN	Attribute	DNN
?	Differential Privacy	n-gram	Sequential/Time Series	w.o. DNN
Cunningham et al. [2021]	Local Differential Privacy	n-gram	Trajectory	w.o. DNN
Du et al. [2023]	Local Differential Privacy	Markov Probabilistic Model	Trajectory	w.o. DNN
He et al. [2015b]	Differential Privacy	Markov Probabilistic Model	Trajectory	w.o. DNN
Wang and Sinnott [2017]	Differential Privacy	Markov Probabilistic Model	Social Media	Trajectory
Gursoy et al. [2018]	Differential Privacy	Markov Probabilistic Model	Trajectory	w.o. DNN
Mir et al. [2013]	Differential Privacy	Distribution Estimation	Location	w.o. DNN
Roy et al. [2016]	Differential Privacy	Distribution Estimation	Trajectory	w.o. DNN
Bindschaedler and Shokri [2016]	Plausible Deniability	Hidden Markov Models	Trajectory	w.o. DNN
Wang et al. [2023]	Differential Privacy	Markov Chain Model	Trajectory	w.o. DNN
Narita et al. [2024]	Differential Privacy	Probabilistic Transform	Trajectory	w.o. DNN
Bindschaedler et al. [2017]	Plausible Deniability	Probabilistic Transform	Attribute	w.o. DNN
Tseng and Wu [2020]	Compressive Privacy	GAN	Image	DNN
Zhang et al. [2018]	Differential Privacy	GAN	Image	DNN
Xie et al. [2018]	Differential Privacy	GAN	Image and EHR	DNN
Xu et al. [2019]	Differential Privacy	GAN	Image	DNN
Liu et al. [2019]	Differential Privacy	GAN	Image	DNN
Triastcyn and Faltings [2020]	Differential Privacy	GAN	Attribute (Tabular) and Graph	DNN
Ge et al. [2020]	Differential Privacy	GAN	Image and EHR	DNN

Table 3: Summary of Some Privacy-Enhancing Techniques in Generative AI for Synthetic Data [1]

11. Evaluation

Evaluating the quality of synthetic data is essential to validate its effectiveness and applicability in practical applications. Key strategies include human evaluation, which relies on expert assessments to judge data quality but is often resource-intensive and may not scale well for high-dimensional datasets. Statistical evaluation offers a quantitative approach by comparing real and synthetic datasets across various metrics, allowing for objective assessments of data fidelity. Additionally, pre-trained machine learning models can serve as discriminators, assessing how closely synthetic data approximates real data, a common technique in Generative Adversarial Networks (GANs) [31]. The"Train on Synthetic, Test on Real" (TSTR) approach evaluates synthetic data by training models on it and measuring performance on real data, thus gauging its utility for downstream tasks. Lastly, application-specific evaluations consider unique domain requirements, such as regulatory compliance and usability, to ensure synthesized data meets specific standards. By combining these methods, researchers can achieve a comprehensive understanding of synthetic data's strengths and limitations, which is pivotal for advancing generation techniques and expanding their applications across fields.

11.1 Human-Based Evaluation

Human evaluation is a fundamental, though often challenging, method to assess the quality of synthetic data [32]. This approach involves gathering feedback from domain experts or general users to judge the data's realism, usability, and similarity to actual data within specific applications. Human evaluation plays a particularly crucial role in tasks where subjective interpretation is essential, such as speech synthesis [33]. where evaluators rate the perceived naturalness and clarity of synthesized voices compared to real human speech in a blind, side-by-side manner [8]. This method allows evaluators to provide insights into subtle nuances that automated metrics might overlook, such as intonation, articulation, and fluidity, which are vital for creating high-quality, user-friendly synthetic voices. Similarly, in computer vision, human judges may assess the accuracy and realism of synthetic images, examining details like texture, lighting, and object consistency, which can be critical for applications in virtual reality and gaming. Despite its advantages, human evaluation has notable limitations. It is resource-intensive, requiring both time and financial investment to gather and analyze opinions from experts or a broad range of users. This method is also subject to variability and potential bias, as human judgments can differ due to individual perceptions, experiences, and interpretation of quality standards. Scalability becomes another hurdle, as this process does not easily extend to evaluating large volumes of high-dimensional data, such as complex image or video datasets, which cannot be fully examined by a human evaluator due to time constraints. High-dimensional synthetic data often contains intricate patterns or attributes that are challenging to assess through visual inspection alone. Moreover, for areas like medical image synthesis or genomic data, human evaluators may lack the ability to validate highly technical details, further limiting the utility of this approach. As a result, while human evaluation provides valuable qualitative insights, it is often best complemented with objective, automated evaluation techniques to obtain a more comprehensive assessment of synthetic data quality and applicability.

11.2 Statistical-Based Evaluation

Statistical difference evaluation is a widely-used strategy to quantitatively assess the quality of synthetic data by comparing statistical metrics between synthetic and real datasets. This approach involves calculating key statistics, such as mean, variance, and correlation, for individual features within both datasets. The closer these statistical properties are, the better the quality and fidelity of the synthetic data. For instance, in electronic health record (EHR) data generation, metrics like the frequency and correlation of medical concepts, as well as patient-level clinical features, are examined to ensure that synthetic data closely mirrors real-world patterns [8]. Smaller statistical differences suggest that the synthetic data has successfully captured the underlying distribution of the real data, making it a valuable proxy for various downstream applications. Advanced techniques such as Support Vector Machines (SVMs) can be utilized to enhance statistical difference evaluation. By training SVMs on synthetic and real datasets, researchers can examine how well the models separate or align these two datasets. In cases where the SVM achieves a high accuracy in differentiating between real and synthetic data, it may indicate notable differences in their distributions. Conversely, if the model struggles to separate them, it suggests that the synthetic data closely approximates the real data distribution. These methods offer a robust, objective means to evaluate similarity, allowing researchers to refine synthetic data generation techniques to achieve better quality and utility across various applications.

11.3 Using Pretrained Models

Using a pre-trained machine learning model to evaluate synthetic data quality provides an automated, robust method for assessing how well the synthetic data approximates real data. In the context of Generative Adversarial Networks (GANs) [3]. this approach leverages the discriminator, a model trained to distinguish between real and synthetic (fake) data, as a quality measure. As the generator improves, it learns to produce data that increasingly"fools" the discriminator, making it difficult for the discriminator to differentiate synthetic data from real data. The discriminator's accuracy or confidence level when evaluating the synthetic data thus serves as an indicator of the generator's success in producing realistic data. A low performance of the discriminator suggests that the synthetic data closely resembles the real data, signifying a high-quality output.

This evaluation strategy is not limited to GANs. Pre-trained machine learning models, such as image classifiers or language models, can also serve this purpose across various types of synthetic data. For example, in synthetic image generation, a pre-trained image classifier can be used to evaluate the synthetic images by measuring how well it classifies them compared to real images. Similarly, for text data, a language model's perplexity on synthetic data relative to real data can provide insights into quality. The strength of this approach lies in its ability to provide automated, task-specific feedback on the realism of synthetic data, making it a versatile evaluation tool across different generative models and domains. This method helps researchers refine generative techniques, ultimately enhancing the realism and applicability of synthetic data in practical settings.

11.4 Train on Synthetic, Test on Real

The ”Train on Synthetic, Test on Real” (TSTR) strategy is a powerful evaluation method for assessing the quality of synthetic data in terms of its utility for machine learning applications. In this approach, models are trained exclusively on synthetic data, then tested on real data to measure their performance in downstream tasks. High performance on real test data implies that the synthetic data effectively captures the essential characteristics and patterns of the real data, making it a viable substitute for training purposes. This approach is particularly useful in scenarios where access to real data is restricted due to privacy or availability concerns, as it enables researchers to assess whether models trained on synthetic data can generalize well to real-world conditions. For example, in, synthetic data is used to train machine-learning models, and their prediction performance is then evaluated on real test data in healthcare applications [34]. This method provides valuable insights into the generalizability of models trained on synthetic datasets, as high TSTR performance across diverse applications—such as classification, regression, or segmentation tasks—indicates that the synthetic data can serve as an effective proxy. Additionally, TSTR enables developers to identify specific aspects where synthetic data may fall short, guiding further improvements in data generation methods to enhance real-world applicability. This strategy thus not only evaluates synthetic data quality but also supports broader adoption of synthetic data in fields where high-quality, representative data is often scarce or sensitive.

12. Future Work

To further advance the field of synthetic data generation, several key areas warrant additional exploration and development. One significant avenue is the capability to generate larger and more diverse datasets. Expanding the capacity to synthesize extensive datasets with high variability would greatly enhance the applicability of synthetic data in machine learning tasks, especially in domains where data scarcity remains a challenge. Moreover, exploring innovative architectures beyond the current models can lead to substantial advancements. Investigating new generative models or enhancing existing ones could improve the quality and diversity of synthetic data. Importantly, demonstrating that these advancements can be achieved using accessible computational resources, such as a personal computer with a well-coded pipeline, would underscore the feasibility of cutting-edge AI developments without the need for extensive infrastructure. This democratization of technology could encourage broader participation in the field and accelerate innovation. Additionally, integrating more robust privacy-preserving techniques into the data generation process remains a critical area for future work. As privacy, concerns continue to grow, developing methods that ensure data utility while rigorously protecting sensitive information is essential. Combining differential privacy mechanisms with generative models could provide stronger guarantees and expand the adoption of synthetic data in sensitive domains.

Finally, applying synthetic data generation techniques to a wider range of applications, including those with complex data types such as time-series, graphs, and multimodal data, would significantly broaden the impact of this research. Tailoring generative models to handle these complex data structures effectively could open new opportunities in various fields, from healthcare to finance, where such data types are prevalent.

13. Conclusion

In conclusion, our framework for synthetic data generation, complemented by an extensive survey of existing methods, has demonstrated its effectiveness in producing high-quality synthetic data across a range of applications. Through this survey, we highlighted the strengths and limitations of various approaches, offering insights into their real-world applicability and potential for enhancing privacy-preserving practices. Our results show that sequence models, in particular, can be effectively utilized to generate large-scale, structured numerical datasets, even in scenarios where original data is limited or subject to strict privacy constraints. By addressing these key limitations and integrating privacy-preserving techniques, our approach not only improves data availability but also ensures the integrity and confidentiality of sensitive information. The scalability and adaptability of our framework, combined with the insights from our survey, position it as a valuable tool for advancing machine learning systems across diverse domains, enabling secure, ethical, and effective synthetic data generation [35-56].

Acknowledgments

The authors would like to express their sincere gratitude to Dr. Samir Mustapha for his invaluable guidance and insightful contributions to the structuring and development of this paper. His support and feedback have been instrumental in enhancing the clarity and organization of our work.

References

1. Lu, Y., Shen, M., Wang, H., Wang, X., van Rechem, C., Fu, T., & Wei, W. (2023). Machine learning for synthetic data generation: a review. *arXiv preprint arXiv:2302.04062.*
2. Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016, October). Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security* (pp. 308-318).
3. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems, 27.*
4. Antoniou, A. (2017). Data Augmentation Generative Adversarial Networks. *arXiv preprint arXiv:1711.04340.*
5. Ring, M., Schlör, D., Landes, D., & Hotho, A. (2019). Flow-based network traffic generation using generative adversarial networks. *Computers & Security, 82*, 156-172.
6. Borji, A. (2019). Pros and cons of GAN evaluation measures. *Computer vision and image understanding, 179,* 41-65.
7. Liang, D., Krishnan, R. G., Hoffman, M. D., & Jebara, T. (2018, April). Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference* (pp. 689-698).
8. Bengio, Y., Ducharme, R., & Vincent, P. (2000). A neural probabilistic language model. *Advances in neural information processing systems, 13.*
9. Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems.*
10. Ioffe, S. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167.*
11. He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision* (pp. 1026-1034).
12. Oord, A. V. D. (2016). WaveNet: A Generative Model for Raw Audio. *arXiv preprint arXiv:1609.03499*.
13. William Cavnar and John Trenkle. N-gram-based text categorization. *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval,* 05 2001.
14. Wu, Y., & Johnson, J. (2021). Rethinking" batch" in batchnorm. *arXiv preprint arXiv:2105.07576.*
15. Sallab, A. E., Sobh, I., Zahran, M., & Shawky, M. (2019). Unsupervised neural sensor models for synthetic lidar data augmentation. *arXiv preprint arXiv:1911.10575*.
16. Marwala, T., Fournier-Tombs, E., & Stinckwich, S. (2023). The use of synthetic data to train ai models: Opportunities and risks for sustainable development. *arXiv preprint arXiv:2309.00652.*
17. Azizi, S., Kornblith, S., Saharia, C., Norouzi, M., & Fleet, D. J. (2023). Synthetic data from diffusion models improves imagenet classification. *arXiv preprint arXiv:2304.08466.*
18. Sergey I. Nikolenko. Synthetic data for deep learning, 2019.

URL https://arxiv.org/abs/ 1909.11512.
19. Mumuni, A., Mumuni, F., & Gerrar, N. K. (2024). A survey of synthetic data augmentation methods in machine vision. *Machine Intelligence Research*, 1-39.
20. Ziyuan Zhao, Kaixin Xu, Shumeng Li, Zeng Zeng, and Cuntai Guan. *MT-UDA: Towards Unsupervised Cross-modality Medical Image Segmentation with Limited Source Labels,* pages 293–303. 09 2021. ISBN 978-3-030-87192-5. doi: 10.1007/978-3-030-87193-2 28.
21. Chen, Y., Li, W., Chen, X., & Gool, L. V. (2019). Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1841-1850).
22. Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. Unsupervised domain adaptation for semantic segmentation with gans. *CoRR*, abs/1711.06969, 2017. URL http://arxiv.org/abs/1711.06969.
23. Sankaranarayanan, S., Balaji, Y., Jain, A., Lim, S. N., & Chellappa, R. (2018). Learning from synthetic data: Addressing domain shift for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3752-3761).
24. Wood, E., Baltrušaitis, T., Hewitt, C., Dziadzio, S., Cashman, T. J., & Shotton, J. (2021). Fake it till you make it: face analysis in the wild using synthetic data alone. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 3681-3691).
25. Werchniak, A., Chicote, R. B., Mishchenko, Y., Droppo, J., Condal, J., Liu, P., & Shah, A. (2021, June). Exploring the application of synthetic audio in training keyword spotters. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 7993-7996). IEEE.
26. Tianfan Fu, Yanmin Qian, Yuan Liu, and Kai Yu. Tandem deep features for text-dependent speaker verification. In *Interspeech 2014,* pages 1327–1331, 2014. doi: 10.21437/Interspeech. 2014-329.
27. Ling, Z. H., Deng, L., & Yu, D. (2013). Modeling spectral envelopes using restricted Boltzmann machines and deep belief networks for statistical parametric speech synthesis. *IEEE transactions on audio, speech, and language processing, 21*(10), 2129-2139.
28. Fazel, A., Yang, W., Liu, Y., Barra-Chicote, R., Meng, Y., Maas, R., & Droppo, J. (2021). Synthasr: Unlocking synthetic data for speech recognition. *arXiv preprint arXiv:2106.07803.*
29. Ambulgekar, S., Malewadikar, S., Garande, R., & Joshi, B. (2021). Next Words Prediction Using Recurrent NeuralNetworks. In *ITM Web of Conferences* (Vol. 40, p. 03034). EDP Sciences.
30. Assefa, S. A., Dervovic, D., Mahfouz, M., Tillman, R. E., Reddy, P., & Veloso, M. (2020, October). Generating synthetic data in finance: opportunities, challenges and pitfalls. In *Proceedings of the First ACM International Conference on AI in Finance* (pp. 1-8).
31. Jordon, J., Yoon, J., & Van Der Schaar, M. (2018, September). PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International conference on learning representations.*
32. Feng, K., Ding, K., Ma, K., Wang, Z., Zhang, Q., & Chen, H. (2024). Sample-Efficient Human Evaluation of Large Language Models via Maximum Discrepancy Competition. *arXiv preprint arXiv:2404.08008*.
33. Donahue, C., McAuley, J., & Puckette, M. (2018). Adversarial audio synthesis. *arXiv preprint arXiv:1802.04208.*
34. Esteban, C., Hyland, S. L., & Rätsch, G. (2017). Real-valued (medical) time series generation with recurrent conditional gans. *arXiv preprint arXiv:1706.02633.*
35. Acs, G., Melis, L., Castelluccia, C., & De Cristofaro, E. (2018). Differentially private mixture of generative neural networks. *IEEE Transactions on Knowledge and Data Engineering, 31*(6), 1109-1121.
36. Bindschaedler, V., & Shokri, R. (2016, May). Synthesizing plausible privacy-preserving location traces. In *2016 IEEE Symposium on Security and Privacy (SP)* (pp. 546-563). IEEE.
37. Bindschaedler, V., Shokri, R., & Gunter, C. A. (2017). Plausible deniability for privacy-preserving data synthesis. *arXiv preprint arXiv:1708.07975.*
38. Cunningham, T., Cormode, G., Ferhatosmanoglu, H., & Srivastava, D. (2021). Real-world trajectory sharing with local differential privacy. *arXiv preprint arXiv:2108.02084.*
39. Du, Y., Hu, Y., Zhang, Z., Fang, Z., Chen, L., Zheng, B., & Gao, Y. (2023). Ldptrace: Locally differentially private trajectory synthesis. *Proceedings of the VLDB Endowment, 16*(8), 1897-1909.
40. Ge, C., Mohapatra, S., He, X., & Ilyas, I. F. (2020). Kamino: Constraint-aware differentially private data synthesis. *arXiv preprint arXiv:2012.15713.*
41. Gursoy, M. E., Liu, L., Truex, S., Yu, L., & Wei, W. (2018, October). Utility-aware synthesis of differentially private and attack-resilient location traces. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security* (pp. 196-211).
42. He, X., Cormode, G., Machanavajjhala, A., Procopiuc, C., & Srivastava, D. (2015). DPT: differentially private trajectory synthesis using hierarchical reference systems. *Proceedings of the VLDB Endowment, 8*(11), 1154-1165.
43. Lee, D., Yu, H., Jiang, X., Rogith, D., Gudala, M., Tejani, M., ... & Xiong, L. (2020). Generating sequential electronic health records using dual adversarial autoencoder. *Journal of the American Medical Informatics Association, 27*(9), 1411-1419.
44. Liu, Y., Peng, J., James, J. Q., & Wu, Y. (2019, December). PPGAN: Privacy-preserving generative adversarial network. In *2019 IEEE 25Th international conference on parallel and distributed systems (ICPADS)* (pp. 985-989). IEEE.
45. Mir, D. J., Isaacman, S., Cáceres, R., Martonosi, M., & Wright, R. N. (2013, October). Dp-where: Differentially private modeling of human mobility. In *2013 IEEE international conference on big data* (pp. 580-588). IEEE.
46. Narita, J., Murakami, T., Hino, H., Nishigaki, M., & Ohki, T. (2024). Synthesizing differentially private location traces including co-locations. *International Journal of Information Security, 23*(1), 389-410.
47. Roy, H., Kantarcioglu, M., & Sweeney, L. (2016). Practical differentially private modeling of human movement data. In *Data and Applications Security and Privacy XXX: 30th Annual IFIP WG 11.3 Conference, DBSec 2016, Trento,*

Italy, July 18-20, 2016. Proceedings 30 (pp. 170-178). Springer International Publishing.
48. Triastcyn, A., & Faltings, B. (2020). Federated generative privacy. *IEEE Intelligent Systems, 35*(4), 50-57.
49. Tseng, B. W., & Wu, P. Y. (2020). Compressive privacy generative adversarial network. *IEEE Transactions on Information Forensics and Security, 15,* 2499-2513.
50. Wang, H., Zhang, Z., Wang, T., He, S., Backes, M., Chen, J., & Zhang, Y. (2023). {PrivTrace}: Differentially Private Trajectory Synthesis by Adaptive Markov Models. In *32nd USENIX Security Symposium (USENIX Security 23)* (pp. 1649-1666).
51. Wang, S., & Sinnott, R. O. (2017). Protecting personal trajectories of social media users through differential privacy. *Computers & Security, 67*, 142-163.
52. Xie, L., Lin, K., Wang, S., Wang, F., & Zhou, J. (2018). Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739.*
53. Xu, C., Ren, J., Zhang, D., Zhang, Y., Qin, Z., & Ren, K. (2019). GANobfuscator: Mitigating information leakage under GAN via differential privacy. *IEEE Transactions on Information Forensics and Security, 14*(9), 2358-2371.
54. Zhang, X., Ji, S., & Wang, T. (2018). Differentially private releasing via deep generative model (technical report). *arXiv preprint arXiv:1801.01594.*